Texas Instrument 99/4A and Myarc 9640 Computers

Spirit of 99

CENTRAL OHIO NINETY NINERS INC.

THE OFFICIAL NEWSLETTER OF THE CENTRAL OHIO NINETY-NINERS INC.

PUBLISHED MONTHLY IN COLUMBUS OHIO

DECEMBER

$1.50 VOL.7 NO.12 DEC 1989

Spirit of 99

THE OFFICIAL NEWSLETTER OF CENTRAL OHIO NINETY-NINERS

Spirit of 99 is published monthly for Central Ohio Ninety-Niners Inc. by C.O. N.N.I. members and is the official news letter of C.O.N.N.I. User Group.

Editorial, advertising and subscription address is:
181 HEISCHMAN AVE
WORTHINGTON,OH 43085

Subscription rate (USA) $20.00 /1 year (12 issues). Foreign subscription rates available upon request. Third class postage paid at Columbus, Ohio.

CHANGE OF ADDRESS: Send both OLD and NEW address to: Subscription address above. WE assume no responsibility for manscripts, programs (tape or disk)not accompanied by return postage. Letters to the Editor become property of Spirit of 99. If published, we reserve the right to edit at our discretion.

OPINIONS EXPRESSED HEREIN ARE THE AUTHORS AND ARE BASED ON VALID DOCUMENTABLE RESEARCH.THEY DO NOT NECESSARILY REFLECT THE OPINIONS OF THE PUBLISHER.

We will not knowingly publish copyright material without the permission of the author and credit due.

All programs published herein are of public domain unless otherwise noted.

Other non-profit user groups may use material from this newsletter only if source and credit is given.

Central Ohio Ninety Niners Inc. is a non profit organization comprised of MEMBERS who own or use the TI99/4A computer and it's related products and have paid a yearly membership fee of $28.00 and whose main objective is the exchange of Educational and Scientific information for the purpose of computer literacy.

C.O.N.N.I. meetings are held the 2nd Saturday of each month at the Martin Janis Senior Center - East Eleventh Ave. at the Ohio State fairgrounds.Meeting time is at 9 am. Meetings are open to the public. Membership dues ($28.00) are payable yearly to C.O.N.N.I. and cover the immediate family of the member. (An application has been placed in this newsletter for your convenience) Please address it to:
EVERETT WADE
179 ERIE ROAD
COLUMBUS, OH 43214

ADVERTISEMENT:
We do accept commercial advertisement at The following rates:

Business Card(2x3.5):	$5.00/issue
1/4 Page:	$25.00
1/2 Page:	$45.00
Full Page:	$75.00

Write this newsletter for other size arrangements.

All ads should be submitted (camera ready) to advertising address above,payment enclosed. Members ads are published at no cost. (Limit of 25 words and must not be commercial please.)

! INDEX !

EDITOR.................JEAN HALL
ASSIST............CAROLE PARKINS

!! OFFICERS !!

* NEW MEETING PLACE *
* CHEMICAL ABSTRACTS *
* 2540 OLENTANGY RD *
****

ANNOUNCEMENTS

Dues are usually paid at or before the March meeting, and are $28 per year for full membership, library and voting privileges, plus the newsletter. You may also pay your dues in two installments if desired: $14 in March and $14 in September. If only the newsletter is desired, then payment is $20 per year. Those who join during other months of the year pay a lesser, pro-rated amount:

Mar---28.00	Apr---25.75	May---23.50	Jun---21.00	Jul---18.75
Aug---16.50	Sep---14.00	Oct---11.25	Nov----9.50	Dec----7.00
Jan----4.75	Feb----2.50			

Fill out an application blank (one on the back of this newsletter), make a check out to C.O.N.N.I. and give it to Everett Wade, the membership registrar, at one of the meetings or mail to him at the following address:

Everett Wade 179 Erie Rd Columbus, OH 43214

MEETING AGENDA ----- SATURDAY 16 DEC 1989

8:30 AM SETUP. COFFEE and DOUGHNUTS
9:00 AM TI-Writer class - Jean Hall
Help for beginners
Games group
DOS S.I.G.
9:45 AM Advanced Art Programs
by Ken Marshall, Jr
D.O.M. available
Raffle tickets available
MICROpendium/Blank disks
for sale
All Libraries open

10:30 AM BUSINESS MEETING

11:00 AM DEMO New offerings from Bud Mills by Chuck Grimes

12:00 PM LUNCH: Bring your own.
Question and Answer session

12:45 PM Classes:
Multiplan

1:30 PM Assembly - Karl Romstedt

2:15 PM Cleanup and departure

ANNOUNCEMENT

WATCH FOR THE LATEST UPDATE FOR TI-TAX BY WILLIAM CHAVANNE

SATURDAY MORNING COFFEE ANYONE?

Call Jim Seitz (875) 5532) to be a host or hostess. SIGN UP IF YOU WANT ANY COFFEE!!

DEC - Everett Wade

WEDNESDAY EVENING MEETING - DEC 27

7:30 PM AT MC DONALD'S CORNER OF CLEVELAND AND AND MAIN IN WESTERVILLE

HOPE TO SEE YOU THERE

FROM THE PRESIDENT'S DESK
by DICK BEERY

With a number of other C.O.N.N.I. members, I recently returned from the Chicago and Milwaukee Faires. Once again, new products have appeared to delight and assist us. TI-Artist Plus, which I reviewed with Ken Marshall in the last issue, is a real knockout. Other new releases include volume II of the Starfleet Technical Drawings, two new disks of Artist Instances, Legends II: The Sequel, Artist Print Shop, Color Card, Color Flyer, Card and Flyer Graphics disks, a disk of Cards for All Occasions, etc. The latter six all are available from Roger Merrit's Comprodine organization. JP Software, formerly Genial Computerware, has a stunningly beautiful, and FAST, version of Chainlink Solitaire, in 100% Assembly language. In the October Micropendium, there appears a review of TIW graphics, a program that apparently lets one create logos for use with TI-Writer in creating letterheads. Paul Scheidmantle has released two disks containing fifty new fonts for PagePro. JP Fonts is a disk of new fonts for TI-Artist. An enhanced PC-Transfer is available, along with a diskful of utilities that allow one to transfer Multiplan Sylk files, among other options. Jerry Stern has a program listing in the same issue of Micropendium for a FILELABEL that works like a charm. The Tigercub has a new CATWRITER that allows one to use much longer program names than those allowed in disk catalogs. The latter also has out an improved and corrected version of DATAWRITER. This program allows anyone to add instructions for the user to any Basic or Extended Basic program easily. A LABELMAKER from Tigercub allows for different fonts and print sizes on each of the five lines of the label, but no graphics. I use this program constantly.

To comment on the Faires: once again, I and many others had a most enjoyable time in both Chicago and Milwaukee. My own particular preference would be to continue to hold the Chicago Faire at the Holidome in Rolling Meadows, although I realize that many problems involving changes in management plagued those working so hard to make this Faire the outstanding success that it was. It is really nice to have all of the activities, including housing, under one roof. We believe it adds greatly to the contact opportunities among participants. So, if it can be so arranged, we hope that the Faire will continue to be presented in this manner. (And that doesn't even include mention of the excellent meal the Holiday staff prepared for the banquet!) Hats off to Sandy Bartels, Marcy Brun, Don Jones and the many others who worked so hard and achieved so much! Our deep and sincere thanks to those in management roles at both faires who were so gracious, generous and tireless in their efforts to make our visit pleasant and profitabe to all. One regrettable glitch at Chicago seemed to be a recurring mismatch between equipment needs of the presenters and that which was provided. I am told that several times a talk was delayed for this reason. I am sure this will be adequately addressed at next year's Faire. I was personally kept busy at the booth and saw only a couple of the demonstrations, one of them being the music demonstration by Bruce Harrison of Harrison Software, and the other a multi-faceted demo by the highly-talented Mike "Frogman" Maksimik. The music of the former, in a paraphrase of the words of Don Jones, is the closest to non-computer music in sound and in programming of nuances of anything yet to be experienced. (Listen to the disk of the Nutcracker Suite, if you don't believe me!). Mike gave us updated information on his F-DOS, showed us how to choose and connect an inexpensive mouse to the 4A, and gave an impressive demonstration of MIDI using a relatively inexpensive keyboard and the RS232 serial port. Both speakers were great, in my opinion! Others in our group reported to me their satisfaction with Gary Bowser's demonstration of his Rambo, a type of "widgit" that allows eight cartridges to be plugged in simultaneously and accessed easily, and the demonstration by Jan Janowski of his methods for producing a portable 4A computer. I apologize to those whom I may have omitted; I either did not personally attend their lecture or did not talk with any friends who did.

It was a pleasure to meet and chat with Barry Boone, Barry Traver, and others who have contributed so much to both the 4A and the Geneve in both their programming efforts and general support. To me, this is one of the great pleasures of attending these two Faires--having the opportunity of meeting personally the "movers and shakers" in the world of our computers.

C.O.N.N.I. had at least twelve in attendance at Chicago, three fewer at Milwaukee. The offerings at Milwaukee may have seemed less extensive than previously, but the participants were, as always, dedicated, loyal and interested. A good show. Thanks to Gene Hitz and all the others of the M'waukee group for their dedicated work and fine achievement. A speaking appearance by Lou Phillips of Myarc was a crowning achievement at this faire, a first for Milwaukee, I believe. C.O.N.N.I. member Ken Marshall, Jr., carried away TWO of the five grand prizes in the raffle, so the trip was even more worthwhile for him and for us.

Some matters of perhaps more local interest: first, on the weekend following the two faires mentioned above, Irwin Hott and Ken Marshall Jr. drove to a shopping mall at Olean, NY to give a demonstration of the use of speech with the TI by the blind. Bill Cook of the sponsoring group opened his home to the two and to Tonka, Irwin's faithful guide dog. There was extensive coverage by the local press, and the mayor of Olean presented Irwin with the key to the city engraved with his name. A delightful experience for all. Congratulations, Irwin! Second, our venture into a new date (third Saturday instead of second) and a new location (Chemical Abstracts building instead of the Martin Janis Center) seemed very pleasing to all who attended our first official, but really our second meeting under the new circumstances. We now have much more ample time available (8:30 AM to 2:30 PM), so classes can be held before and after the general meeting. Many brought in their computers (three were Geneves, and one the excellent portable 4A that Bud Wright recently bought from Curt Borders, its creator), and we have heard many favorable comments on the new setup. If you have not already done so, please join us and see for yourself! But, BRING YOUR LUNCH, as food is not available on the premises, though two microwave ovens are. We hope to explore even further the greater flexibility offered at Chemical Abstracts, so as to make your meetings even more profitable and enjoyable. Third, we are once again exhibiting at a multi-computer show. C.O.N.N.I. will have a table at the Silicon Express show on Sunday, December 17th, at the Aladdin Shrine on Stelzer Road, from 10AM to 3PM. See you there! Those who have a second or third computer that is non-TI may find good software and hardware for these also.

We hope to see you: one, at our regular Saturday and Wednesday meetings; two, at next year's Chicago, Milwaukee and Lima fairs; and three, at the Silicon Express show already mentioned. Hope your Thanksgiving holiday was a great one, and that your Christmas and New Years celebrations will be unequaled. We will not be publishing a regular newsletter in January, so my next article will appear in the February issue. Till then!

MINUTES FOR WEDNESDAY,
NOV. 22,1989.

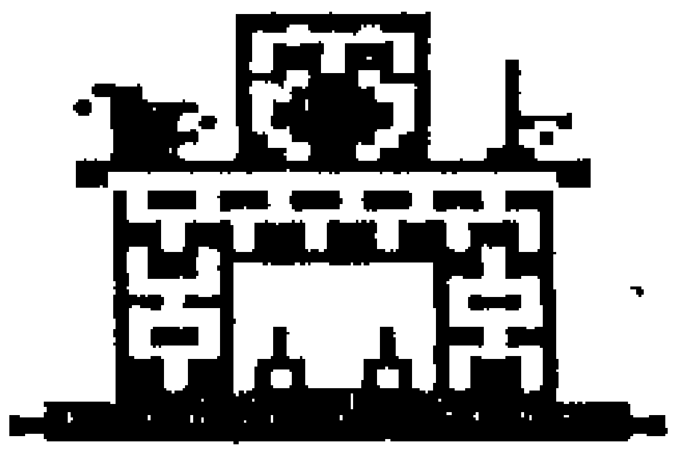
The meeting utilized most of the same themes and topics as that of the preceding Saturday. Further discussion took place concerning possible upgrades for the club's system. Chuck Grimes is heading the committee, and funding will come from the Library Fund, not the general fund. The usual Question-and -Answer period was greatly extended, and included a variety of topics. Many of those who attended the Chicago and Milwaukee faires gave their impressions of these events. Chuck announced that Asgard Software is now offering seven more Infocom games. All seven will retail for $14.95 each, plus sh. Chuck also informed us that the next stage in the development of the P-Gram card available from Bud Mills Services will be an expansion from 80K to 128K, together with the capability of bank-switching between pages of cartridges.

Chuck also provided all the demonstrations, with very little time to prepare them, as he was asked to do so on the previous Saturday. Included were the Nutcracker Suite (Harrison Software); Chainlink Solitaire(JP Software); Mike Dodd's new Identidisk; Gram-Kracker; and a P-Gram card (Bud Mills Services). Other items that would have been presented were not, owing to the club's system having only SSSD drives at this time.

The November Disk-of-the-Month, a freebie, was distributed, and copies of the October Micropendium, blank disks and memberships were available for purchase.

The participants who stayed until the end of the session left, as usual reluctantly, somewhat after the 11PM closing time.

Respectfully submitted,
Dick Beery and Jean Hall
For Charles Osment, who could not be present.



AND THE PRICE IS RIGHT!

A Review of Some Collections
by Dick Beery

Since its inception, our club has had as a member Jim Peterson, a renowned programmer. I felt it was time to give recognition to a recent expansion of his offerings, since many of those who are not in our group but who receive and read this newsletter might wish to be informed about this. Actually, Bill Gaskill "scooped" me in his Four-A/Talk column appearing in the November 1989 issue of The Front Ranger). I have personally purchased a number of items from the new catalog and, following almost every meeting, several of our members do the same. I have greatly enjoyed the ones I have bought, and it seems that all those I have talked with feel the same way. I hope that this review will enable many more people to share the pleasure we have experienced.

The "Tigercub" (Jim Peterson) has, in addition to marketing his own wonderful and prolific output of software for the 4A, prepared a collection of disks of Public Domain and Fairware software at the incredibly low price of $1.50 postpaid for each SSSD disk. The disks are arranged by categories such as Christmas music, some with graphics, pre-school educational programming, assembly-language programs, utilities, country music, and others too numerous to mention.

Historically, in the early years of the 4A, a group known as the International Users' Group sold PD software at three dollars a program. This group was followed by one designated as the 994A National Assistance Group, with the same type of output and price. Most recently, the highly-respected Dr. Guy Steffan-Romano sold DSSD disks from his Amnion library at five dollars per disk.

The Tigercub's PD catalog, which costs only one dollar, refundable with the first purchase, lists over three hundred disks of what Jim feels to be the great majority of the "most worthwhile Public Domain and Fairware offerings in the USA, England, Australia, Canada and Belgium." A number of fairware authors have declared publicly and in print their dissatisfaction with companies that sell their products without permission. The Tigercub has expressly sought and received permission from most fairware authors to include their offerings in his catalog, including J.Peter Hoddie, Clint Pulley, and many others. When permission was not forthcoming, the item was not included in the collection. A unique permission was granted by Bruce Harrison of Harrison Software for Jim to include his copyrighted software in the catalog.

An easy-to-use loader that provides the full program name, rather than the disk filename, is on each disk. In many instances, instructions have been included, and all bugs that have been identified have been corrected.

Recent programs written by Jim himself include his CATWRITER, a disk catalog program that permits the listing of a complete program name instead of the more limited filename (this program is available as a type-in in the October 1989 Micropendium or from the author), and DATAWRITER, a program that permits even a relatively inexperienced person to include help files in Basic or Extended Basic programs.

Address your inquiries and/or requests for catalogs to:

TI-PD
156 Collingwood Avenue
Whitehall, OH 43213.

NOTICE

CONNI USER GROUP ANNOUNCES NEWSLETTER SUBSCRIPTION CHANGES

NEWSLETTER ONLY #15 YEAR

NEWSLETTER WITH DISK OF THE MONTH #30 A YEAR
MINIMUM 10 NEWSLETTERS
AND 10 DISK OF THE MONTH
SOME MONTHS 2 FLIPPIES
FOR OUT OF STATERS

FOREIGN
#36 A YEAR

PUTTING IT ALL TOGETHER No. _

by Jim Peterson

The hard part of learning to program is not in learning what the various commands do - it is in learning to put them together to do what you want them to do!

Key in this little program and run it to see what it does, then read the explanation of how it does it.

```
100 DISPLAY AT(3,11)ERASE AL
L:"SPELLIT" !by Jim Peterson
110 DATA HIPPOPOTAMUS,CRITIQ
UE,KHAKI,IRIDESCENT,ARCHAIC,
PNEUMONIA
120 !add as many DATA statem
ents as you want
130 FOR CH=97 TO 122 :: CALL
 CHARPAT(CH-32,CH$):: CALL C
HAR(CH,CH$):: NEXT CH :: CAL
L COLOR(9,8,2,10,8,2,11,8,2,
12,8,2)
140 DATA END
150 READ M$ :: T=100 :: IF M
$="END" THEN CALL CLEAR :: S
TOP
160 GOSUB 230 :: ACCEPT AT(1
2,1)SIZE(-28)BEEP:Q$
170 IF Q$=M$ THEN CALL SOUND
(100,392,5):: CALL SOUND(200
,523,5):: DISPLAY AT(12,1):"
" :: GOTO 150
180 FOR J=1 TO LEN(Q$):: IF
SEG$(Q$,J,1)=SEG$(M$,J,1)THE
N 210
190 DISPLAY AT(12,J):CHR$(AS
C(SEG$(Q$,J,1))+32);
200 T=T+50 :: IF LEN(Q$)=LEN
(M$)THEN GOSUB 230 :: GOTO 2
10 ELSE DISPLAY AT(12,J+1):"
 " :: J=LEN(Q$):: GOTO 160
210 NEXT J
220 T=T+50 :: GOTO 160
230 DISPLAY AT(10,1):M$ :: F
OR D=1 TO T :: NEXT D :: DIS
PLAY AT(10,1):"" :: RETURN
```

Line 100 erases all the trash from the screen and prints the title centered on line 3. The screen is 28 characters wide. SPELLIT contains 7 characters. 28 minus 7 divided by 2 is 10.5, so center the title at column 11. Put in as many lines of words in DATA as you want.

The lower case characters "a" through "z" are ASCII 97 to 122. The upper case are just 32 below that, ASCII 65 to 90. CALL CHARPAT to get the hex pattern identifier of each upper case letter in CH$, then CALL CHAR to reidentify the corresponding lower case letter to that pattern. The lower case letters are in sets 9 to 12, so color them in the reverse of the normal screen, cyan on black.

The dummy data END in line 140, and the statement in line 150, causes the program to stop without crashing when it runs out of words, regardless of how many you put in.

Line 150 reads each word from DATA one after another and sets the initial time to display it at 100 milliseconds.

Line 160 jumps to line 230 to display the word at line 10 column 1, wait for the set time, then erase it by displaying a null string which erases the line. Then it signals with a beep and cursor that it is waiting for your spelling, Q$, at line 12 column 1.

Line 170 checks whether your spelling is the same as the word M$. If so, it sounds two notes, displays a null string to erase the word and goes back for the next.

If not correct, line 180 starts a loop for the number of letters, LEN(Q$), in your spelling and compares each letter with the letter in the same position in M$. If the same, it jumps to 210 to check the next letter.

But if incorrect, line 190 displays at that point the character of the ASCII 32 higher, which is the same letter in inverted colors.

Line 200 increments the flashing time by 50, then checks to see if the word you spelled is the same length as the correct word. If so, it goes to 230 to flash the correct word, then continues checking letters. When finished, line 220 increments flash time and sends you back to try again. The -28 size in the ACCEPT statement prevents the misspelled word from being erased.

If your spelling has a different number of letters, the first error probably has caused all subsequent letters to be in the wrong position. They would all be marked as wrong, so a null string is displayed to erase the rest of the word. The statement J=LEN(Q$) clears the loop in computer memory to avoid the possibility of a MEMORY FULL error. Then you are prompted to try spelling the word again.

Implanting Assembly in Extended BASIC

by Ross Mudie

(Thanks to Ross Mudie and the Tshug UG

This article is a follow on to the technique described by George Meldrum in the November 1987 TND magazine, which shows how to implant single file programs. This is NOT a beginner's article.

The program chosen is for the example is Sewer Mania. This program has an additional problem of intermittently failing to recognise the Speech Synthesizer however this problem is NOT addressed in this article.

Sewer Mania is in 4 Memory Image files with headers as follows:

SEWERMANIA1	SEWERMANIA2	SEWERMANIA3	SEWERMANIA4
FFFF2000A000	FFFF2000BFFA	FFFF1A00DFF4	00000E002800
Code length			
1FFA	1FFA	19FA	0DFA
Ultimate Load address			
A000 to BFF9	BFFA to DFF3	DFF4 to F9EE	2800 to 35FA
Implanted start address			
DEF6	DFEE	E5EE	F1EE
X/B line numbers ie 8330 and 8332			
DEF5	DFED	E5ED	F1ED

Other memory allocations to be considered.

Assembly Loader FEF0 to FFE7 (not fully using the space).
Loader Work Space 2600 to 261F
Character moving buffer 3600 to 390F
X/B INIT routines 2000 to 24F3
Space for relocation of code if required.
3910 to 3FFF (Not needed for Sewer Mania, but included to assist with facing the problem in other programs.)

Part 1 has the assembly file commencing at FEF0 thus the implanted assembly must reside between DEF6 and FEF0.

Parts 2, 3 and 4 have no assembly, thus the top for these is FFE8 rather than FEF0 which was used for *part 1. The actual top is one byte* less than this, because these addresses are the start of the next code area. Hence the actual tops are FEEF and FFE7. However use FEF0 and FEF8 to get your maths correct.

The Extended BASIC lines follow for the headers of the four parts. It is only necessary to tell X/B that it has no extension memory once with the CALL LOAD(-31868,0,0) in the first part. The 4 X/B headers can be saved as merge files and merged in as necessary. These are saved out of X/B using SAVE DSK1.M1,MERGE etc. I use M1 for part 1, M2, M3, and M4 for parts 2, 3 and 4.

```
100 REM SAVE DSK1.SEWRMAN1XB
110 CALL INIT
120 ON ERROR 190
130 DISPLAY ERASE ALL:"LOADING PART 2"
140 CALL PEEK(-31888,V):: IF V<63 THEN 170
150 DISPLAY :RPT$("¯",28);"* IGNORE REWIND TAPE
    MESSAGE";RPT$("¯",28)
160 CALL LOAD(-31868,0,0):: RUN "CS1"
170 DISPLAY :"DSK1.SEWRMAN2XB FILE"
180 CALL LOAD(-31868,0,0):: RUN "DSK1.SEWRMAN2XB"
190 CALL LOAD(-31868,255,231)

100 REM SAVE DSK1.SEWRMAN2XB
110 CALL LOAD(8196,254,240)
120 ON ERROR 190
125 CALL LINK("SEWER2")
130 DISPLAY ERASE ALL:"LOADING PART 3"
140 CALL PEEK(-31888,V):: IF V<63 THEN 170
150 DISPLAY :RPT$("¯",28);"* IGNORE REWIND TAPE
    MESSAGE";RPT$("¯",28)
160 RUN "CS1"
170 DISPLAY :"DSK1.SEWRMAN3XB FILE"
180 RUN "DSK1.SEWRMAN3XB"
190 CALL LOAD(-31868,255,231)

100 REM SAVE DSK1.SEWRMAN3XB
110 CALL LOAD(8196,254,248)
120 ON ERROR 190
125 CALL LINK("SEWER3")
130 DISPLAY ERASE ALL:"LOADING PART 4"
140 CALL PEEK(-31888,V):: IF V<63 THEN 170
150 DISPLAY :RPT$("¯",28);"* IGNORE REWIND TAPE
    MESSAGE";RPT$("¯",28)
160 RUN "CS1"
170 DISPLAY :"DSK1.SEWRMAN4XB FILE"
180 RUN "DSK1.SEWRMAN4XB"
190 CALL LOAD(-31868,255,231)

100 REM SAVE DSK1.SEWRMAN4XB
110 CALL LOAD(8196,255,0)
120 CALL LINK("SEWER4")
```

Order of doing implanting.

1. Load DEBUG.
2. OLD the part 1 memory image file into memory.*(I/O Error 50 must be received)*
3. CALL LINK("DEBUG")
4. Inspect VDP memory to find size etc eg, M0975V,097F
5. Move the code as necessary, eg. N097BV,DEF6,1FFA
6. Change the X/B line number pointers, eg, M8330 <enter> DEF5 <space> DEF5 <enter>
7. Quit out of Debug, eg, Q <enter>.
8. *Merge X/B header, eg, MERGE DSK1.M1*
9. Load the already assembled object file containing the corect addresses, eg, CALL LOAD("DSK1.MO")
10. Save the implanted program file, eg, SAVE DSK1.SEWRMAN1XB
11. Go to step 2 for the next part or end.

It is *also possible* to save the X/B header with the assembly game program but without the loader. Later when the loader has been edited to provide all the required addresses and assembled, follow the following order.

1. CALL INIT
2. OLD DSK1.SEWRMAN1XB
3. CALL LOAD("DSK1.MO")
4. SAVE DSK1.SEWRMAN1XB

The order of 1 and 2 may be reversed. 3 must follow 1 and 2 so that the code from the object file will overwrite the memory space.

How to tell if the loader is too big.

The assembly loader program has some free space between its end and the end of the space allocated. If you need to add extra code to relocate code etc you may *end* up past FFE7. If this happens then Extended BASIC will overwrite your code and your loader code *will fail* to operate. The way to check where your code will reside is to get a LIST file at assembly. If the printer is on PIO then at the LIST prompt type in PIO. noting that the period is required after the device name. The options required will be RL for Registers prefixed by R and L to produce a List file. Since the AORG signifies absolute code, then column 2 of the list *file will show the addresses in memory and column* 3 will contain the hex machine code. The last address in your code must not exceed FFE6. If your code is too big then it must be made more compact or the AORG must be moved down. If this occurs then the implanted code must be moved down and the LINK addresses changed in the X/B to match the new assembly addresses.

What to do if the program overwrites the loader.

In the example program the part 3 file did not overwrite the area in which the loader was placed. The file at DFF4 is only 19FA bytes long and ends at F9EE. Larger files can overwrite the area FEF0 up and destroy the loader. When this happens it is necessary to move the remaining part of the loader before this happens into free space, then branch into the start of the relocated code. Care must also be taken to avoid *placing the program over your own work space or* using buffer space for the character definition moves which overwrites the program.

To relocate, here is an example of suitable code:

```
LI R2,MOVFIN Address after end of code being
                    moved
LI R0,MOVST  Adress of start of code being moved
S  R0,R2     Subtract to find size
LI R1,>3910  Where to put it, after chars buffer
                    space
```

Con't on page 14

I GOOFED!

Jim Peterson

Well,I really blew it with that DATAWRITER in Tips #58. It worked OK on the programs I used it on, but I should have realized that it won't work with any program that has subprograms, because you can't put anything after a subprogram. And it won't work with a program that reads data without restoring it, because it will try to read beyond the last data item. I don't know why I didn't put the data up front - guess I was too concerned about overwriting something when merging in the routine. I also failed to give the option to skip instructions, so let's start over from scratch. First key this in -

```
1 DISPLAY AT(12,1)ERASE ALL:
"SKIP INSTRUCTIONS? Y" :: AC
CEPT AT(12,20)SIZE(-1)VALIDA
TE("YNyn"):@Q$ :: IF @Q$="Y"
 OR @Q$="y" THEN 8
2 DISPLAY AT(24,5)ERASE ALL:
"PRESS ANY KEY"
3 RESTORE 30721
4 REM
5 FOR J@=1 TO T@ :: READ @$
:: DISPLAY AT(J@,1):@$:" "
6 CALL KEY(0,K@,S@):: IF S@=
0 THEN 6
7 NEXT J@
8 DATA 0
9 RESTORE 8 :: READ N
10 REM
```

Save it by -
SAVE DSK1.D/MERGE,MERGE
Then key this in -

```
100 OPEN #1:"DSK1.D/MERGE",V
ARIABLE 163,INPUT :: OPEN #2
:"DSK1.D/MERGE2",VARIABLE 16
3,OUTPUT :: L=129 :: FOR J=1
 TO 10
110 LINPUT #1:M$ :: PRINT #2
:CHR$(0)&CHR$(L+J)&CHR$(156)
&CHR$(253)&CHR$(200)&CHR$(1)
&"1"&CHR$(181)&CHR$(199)&CHR
$(LEN(M$))&M$&CHR$(0):: NEXT
 J
120 CLOSE #1 :: PRINT #2:CHR
$(255)&CHR$(255):: CLOSE #2
```

Run it to convert D/MERGE into a merge format file D/MERGE2 on DSK1. Then key this in. Don't change line numbers.

```
100 CALL CLEAR :: OPEN #1:"D
SK1.@DATA",VARIABLE 163,OUTP
UT :: DEF L$(X)=CHR$(120)&CH
R$(X)
105 PRINT #1:L$(X)&CHR$(161)
&CHR$(200)&CHR$(6)&"@DUMMY"&
CHR$(0)
110 L=L+1 :: X=X+1 :: ACCEPT
 AT(L,0):M$ :: IF L=24 THEN
CALL CLEAR :: L=0
120 IF M$<>"END" AND M$<>"en
d" THEN PRINT #1:L$(X)&CHR$(
147)&CHR$(199)&CHR$(LEN(M$))
&M$&CHR$(0):: GOTO 110
130 REM
140 PRINT #1:CHR$(0)&CHR$(4)
&"T@"&CHR$(190)&CHR$(200)&CH
R$(LEN(STR$(X-1)))&STR$(X-1)
&CHR$(0)
141 PRINT #1:L$(X)&CHR$(168)
&CHR$(0)
150 PRINT #1:CHR$(255)&CHR$(
255):: CLOSE #1
```

Enter MERGE DSK1.D/MERGE2 to merge in that file. SAVE the program as DATAWRITER. Then RUN it and try it out by using it to write itself some instructions. Answer the prompts with -

DATAWRITER V1.2
by Jim Peterson

To be used to add instructions to programs.

Type the instructions and format them, centered or hyphenated or right-adjusted just as you want them to appear on screen, and enter each line. They will be written to a D/V163 file named @DATA. When finished, enter END.

Then enter NEW, then MERGE DSK1.@DATA, and RUN to see if everything is OK. If so, load the program needing instructions, make sure its lowest line number is more than 10 and the highest is less than 30721, and enter MERGE DSK1.@DATA.

And enter END, then OLD DSK1.DATAWRITER, then MERGE DSK1.@DATA.

A VICIOUS ERROR!!!!!!!!

Your Editor GOOFED

Jean Hall

If you keyed in the program VICIOUS, that appeared in the Nov 1989 issue of SPIRIT of 99 and discovered that the down arrow error key would not work, it was because I made a few typos. The following lines need to be corrected:

Line 300 was corrected by Jim Peterson, so the 1's will disappear as you cover them and now you will obtain a score at the end of your 10 trys.

Line 300, change),R) to),X)

Line 300, change R<>108 to X<>49

My typos:

Line 330, change DE;SPRITE to DELSPRITE

Line 330, change I/[to I=[

Line 400, change the last C=4 to C=-4

Sorry bout that. I hope you did not get to frustrated when you tried to play this game. My apologies to the West Penn UG and any TI users that took the time to key in this program.

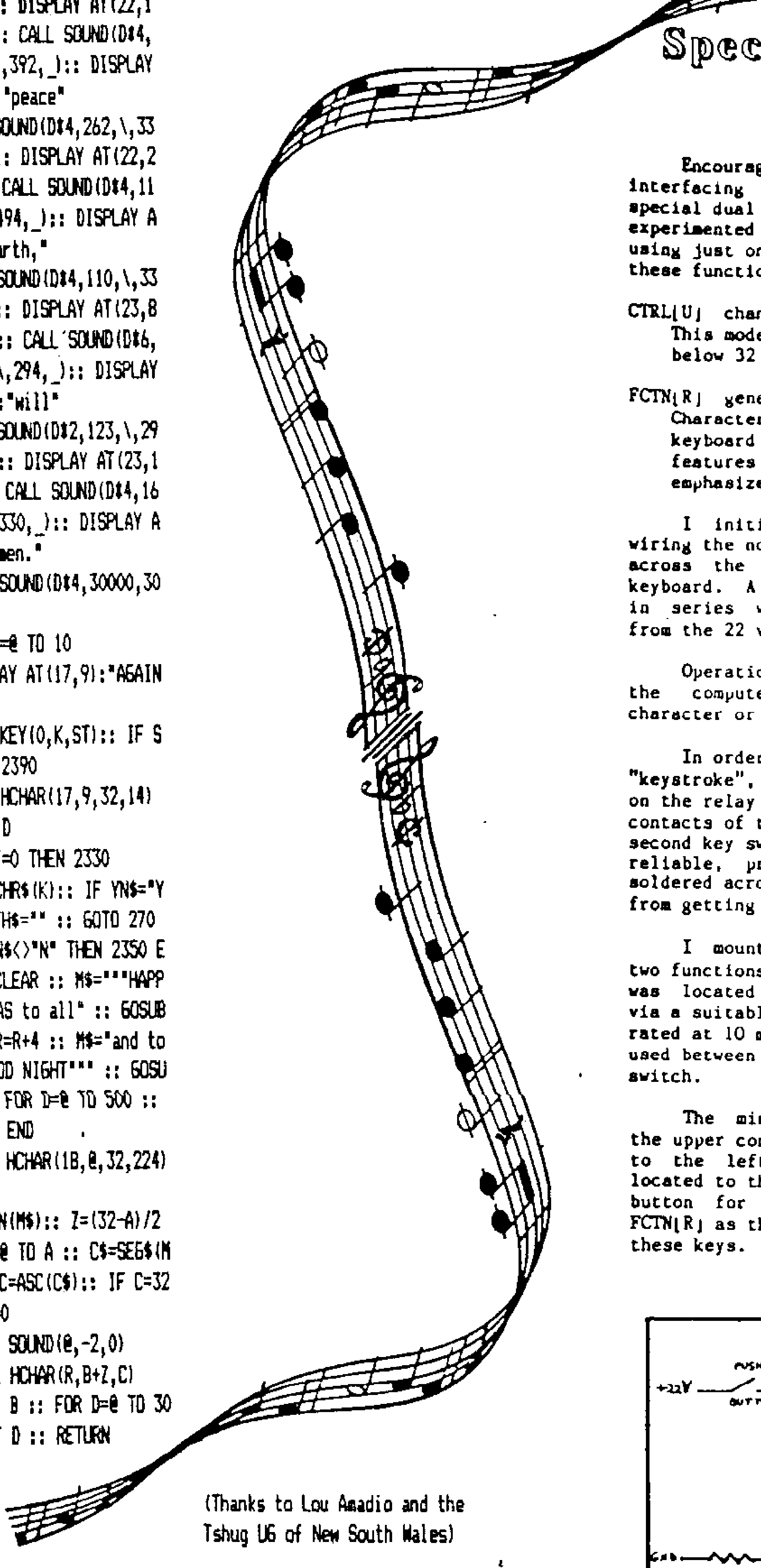
A CHRISTMAS SING-A-LONG

By Carl G. Schwerin

(A type in program. You may *obtain* this *program from* the C.O.N.N.I. library or download from the SPIRIT of 99 BBS, if you do not want to take the time to key in this very long program.)

```
100 !---------------------!
110 !CHRISTMAS SING-A-LONG!
120 ! By Carl G. Schwerin !
130 !    Version 1XB      !
140 ! 32K Memory Expansion!
150 !---------------------!
160 GOTO 180 :: CALL CHAR ::
 CALL CLEAR :: CALL HCHAR ::
 CALL VCHAR :: :: CALL SCREE
N :: CALL CHARSET :: CALL CO
LOR :: CALL KEY :: CALL SOUN
D
170 A,B,C,D,I,K,P,R,SC,ST,Z,
@,\,_ :: A$,B$,BOTH$,C$,D$,F
N$,L$,LN$,M$,SN$,YN$ :: !@P-
180 @=1 :: \=12 :: _=10 :: C
ALL CLEAR :: CALL SCREEN(9):
: A$=RPT$("-",28):: D$="abcd
efghijklmnopqrstuvwzxy"
190 L$="0000FF" :: CALL CHAR
(47,L$):: L$=RPT$(L$,12):: F
OR I=0 TO 12 :: CALL COLOR(I
,16,7):: NEXT I
200 DISPLAY AT(5,2):A$: :TAB
(5);"^ Christmas Scene ^": :
TAB(7);"^ With Carols ^": :A
$
210 DISPLAY AT(13,13):"By":
:TAB(7);"Carl G. Schwerin" :
: CALL HCHAR(16,9,47,16):: F
OR D=@ TO 200 :: NEXT D :: C
ALL CLEAR :: CALL SCREEN(3)
220 DISPLAY AT(2,6):"Combine
d letters":"    in names MUS
T NOT":"   exceed 26 charact
ers."
230 DISPLAY AT(6,@):"Your fi
rst name, Please:" :: ACCEPT
 AT(7,5)VALIDATE(UALPHA,D$)B
EEP SIZE(-25):FN$
240 DISPLAY AT(10,@):"Your s
pouses first name:": :"(If n
one, press enter)" :: ACCEPT
 AT(13,5)VALIDATE(UALPHA,D$)
BEEP SIZE(-25):SN$
250 DISPLAY AT(16,@):"Last n
ame:" :: ACCEPT AT(17,5)VALI
DATE(UALPHA,"'",D$)BEEP SIZE
(-25):LN$ :: IF LEN(FN$&SN$&
LN$)<=26 THEN 270
260 DISPLAY AT(22,@):"Exceed
ed 26 Characters" :: GOTO 23
0
270 CALL CLEAR :: CALL CHARS
ET :: R=17
280 FOR I=0 TO 12 :: CALL CO
LOR(I,16,9):: NEXT I
290 DISPLAY AT(8,@):"1-Trans
parent  9-Med.Red    2-Black
        10-Lt.Red    3-Med.G
reen   11-Dark Yellow4-Lt.Gr
een    12-Lt.Yellow"
300 DISPLAY AT(11,@):"5-Dark
.Blue   13-Dark Green 6-Lt.B
lue     14-Magenta    7-Dark
 Red    15-Gray       8-Cyan
        16-White"
310 DISPLAY AT(18,@):"Backgr
ound Color (1-16)" :: ACCEPT
 AT(18,25)VALIDATE(DIGIT)SIZ
E(-2):SC
320 IF SC<@ OR SC>16 THEN 31
0 ELSE CALL CLEAR
330 CALL SCREEN(SC)
340 IF SC<15 THEN CALL COLOR
(@,16,@,2,16,@,3,10,7,4,7,@,
5,16,@,6,16,@,7,16,@,8,16,@,
9,16,@,10,16,@)
350 IF SC<15 THEN CALL COLOR
(11,16,@,12,16,@,13,4,@,14,4
,@)
360 M$="MERRY CHRISTMAS" ::
GOSUB 2420 :: FOR D=@ TO 10
:: PRINT :: NEXT D
370 R=R+3 :: M$="from" :: GO
SUB 2420
380 IF SN$="" THEN M$=FN$&"
"&LN$ :: R=R+@ :: GOSUB 2420
 :: GOTO 400
390 R=R+@ :: M$=FN$&" and "&
SN$&" "&LN$ :: GOSUB 2420
400 R=R+@ :: M$="Happy New Y
ear" :: GOSUB 2420
410 IF SC>14 THEN CALL COLOR
(2,13,@,3,10,7,4,9,@,5,13,@,
6,13,@,7,13,@,8,13,@,9,13,@)
420 IF SC>14 THEN CALL COLOR
(10,13,@,11,13,@,12,13,@,13,
3,@,14,3,@)
430 IF SC=4 THEN CALL COLOR(
13,13,@,14,13,@)ELSE IF SC=1
3 THEN CALL COLOR(13,5,@,14,
5,@)ELSE IF SC=7 THEN CALL C
OLOR(4,10,@)ELSE IF SC>14 TH
EN CALL COLOR(@,11,@)
440 IF SC=11 OR SC=12 THEN C
ALL COLOR(@,7,@,2,7,@,5,7,@,
6,7,@,7,7,@,8,7,@,9,7,@,10,7
,@,11,7,@,12,7,@)
450 IF SC=@ OR SC=3 OR SC=5
OR SC=7 OR SC=9 OR SC=12 THE
N CALL COLOR(3,15,16)
460 !-Fireplace
470 CALL CHAR(40,"0101010101
010101808080808080800000000
0000FFFFFF")
480 CALL CHAR(48,"FF81818181
8181FF80808080808080B0101010
1010101010")
490 !-Tree
500 CALL CHAR(128,"010303070
707070F80C0C0E0E0E0E0F00F0F0
F1F1F1F1F3FF0F0F8F8F8F8F8FC"
)
510 CALL CHAR(132,"3F3F7F7F7
F7FFFFFFCFCFEFEFEFEFFFF00000
101030303030000808080C0C0C0C0"
)
520 CALL CHAR(136,"0707070F0
F0F0F1FE0E0E0F0F0F0F0F81F1F1
F3F3F3F3F7FF8F8F8FCFCFCFCFE"
)
530 CALL CHAR(140,"7F7F7F7F7
FFFFFFFFEFEFEFEFEFFFFFFFFFFF
FFFFFFFFFFF")
540 !-Wreath
550 CALL CHAR(56,"000001070F
1F3F3C000080E0F0F87C3C",43,"
3C3E1F0F0F010000",47,"3C7CF8
F0E0B00000")
560 !-Star
570 CALL CHAR(35,"0101030F07
03060C8080C0F0E0C06030010103
0F0703060C8080C0F0E0C06030")
580 !-Flame
590 CALL CHAR(60,"0101010303
031313000000222222222213131F
37373F7F7FE2E6F6FEFEFEFEFE")
600 !-Fireplace
610 CALL HCHAR(14,6,48,10)::
 CALL HCHAR(15,6,50,10):: CA
LL HCHAR(16,6,48,10):: CALL
HCHAR(17,6,50,3):: CALL HCHA
R(17,13,50,3)
620 CALL HCHAR(18,6,48,3)::
CALL HCHAR(18,13,48,3):: CAL
L HCHAR(19,6,50,3):: CALL HC
HAR(19,13,50,3):: CALL VCHAR
(14,5,40,6)
630 CALL VCHAR(14,16,41,6)::
 CALL HCHAR(13,5,42,12):: CA
LL VCHAR(17,9,41,3):: CALL V
CHAR(17,12,40,3)
640 !-Tree
650 CALL HCHAR(13,22,128)::
CALL HCHAR(13,23,129):: CALL
 HCHAR(14,22,130):: CALL HCH
AR(14,23,131)
660 CALL HCHAR(15,22,132)::
CALL HCHAR(15,23,133):: CALL
 HCHAR(16,21,134):: CALL HCH
AR(16,24,135)
670 CALL VCHAR(16,22,142,4):
: CALL VCHAR(16,23,142,4)::
CALL HCHAR(17,21,136):: CALL
 HCHAR(17,24,137)
680 CALL HCHAR(18,21,138)::
CALL HCHAR(18,24,139):: CALL
 HCHAR(19,21,140):: CALL HCH
AR(19,24,141)
690 !-Wreath
700 CALL HCHAR(9,10,56):: CA
LL HCHAR(9,11,57):: CALL HCH
AR(10,10,43):: CALL HCHAR(10
,11,47)
710 FOR D=@ TO 35
720 !-Star
730 CALL HCHAR(12,22,32):: C
ALL HCHAR(12,23,32):: CALL H
CHAR(12,22,32):: CALL HCHAR(
12,23,32)
740 CALL HCHAR(12,22,35):: C
ALL HCHAR(12,23,36):: CALL H
CHAR(12,22,37):: CALL HCHAR(
12,23,38)
750 !-Flame
760 CALL HCHAR(18,10,32):: C
ALL HCHAR(18,11,32):: CALL H
CHAR(19,10,32):: CALL HCHAR(
19,11,32)
770 CALL HCHAR(19,11,63):: C
ALL HCHAR(19,10,62):: CALL H
CHAR(18,11,61):: CALL HCHAR(
18,10,60)
780 NEXT D
790 FOR D=1 TO 6 :: PRINT ::
 NEXT D
800 DISPLAY AT(18,@):"PRESS:
[I] Heard the Bells":TAB(7);
"[S]ilent Night":TAB(7);"[B]
oth Carols"
810 CALL KEY(0,K,ST):: IF ST
=0 THEN 810
820 IF K=73 THEN 1630 ELSE I
F K=83 THEN 850 ELSE IF K=66
 THEN 830 ELSE 810
830 B$="YES"
840 !-Silent Night
850 R=R-5 :: GOSUB 2410 :: M
$="Silent Night" :: GOSUB 24
20 :: R=R+@ :: M$="Joseph Mo
hr     Franz Gruber" :: GOSU
B 2420 :: GOSUB 2410
860 D=280 :: CALL SOUND(D*3,
131,\,330,\,392,_):: DISPLAY
 AT(18,@):"Si" :: CALL SOUND
(D,262,\,349,\,440,_)
870 CALL SOUND(D*2,262,\,330
,\,392,_):: DISPLAY AT(18,3)
[illegible]
[illegible]
[illegible](18,6)[illegible]
880 CALL SOUND(D*3,131,\,330
,\,392,_):: DISPLAY AT(18,15
):"Ho" :: CALL SOUND(D,262,\
,349,\,440,_)
890 CALL SOUND(D*2,262,\,330
,\,392,_):: DISPLAY AT(18,17
):"ly" :: CALL SOUND(D*4,196
,\,262,\,330,_):: DISPLAY AT
(18,20):"night,"
900 CALL SOUND(D*4,196,\,349
,\,587,_):: DISPLAY AT(19,@)
:"All" :: CALL SOUND(D*2,196
,\,349,\,587,_):: DISPLAY AT
(19,5):"is"
910 CALL SOUND(D*6,196,\,349
,\,494,_):: DISPLAY AT(19,8)
:"calm," :: CALL SOUND(D*4,1
31,\,330,\,523,_):: DISPLAY
AT(19,14):"all"
920 CALL SOUND(D*2,131,\,330
,\,523,_):: DISPLAY AT(19,18
):"is" :: CALL SOUND(D*6,131
,\,330,\,392,_):: DISPLAY AT
(19,21):"bright,"
930 CALL SOUND(D*4,175,\,349
,\,440,_):: DISPLAY AT(20,@)
:"'Round" :: CALL SOUND(D*2,
175,\,349,\,440,_):: DISPLAY
 AT(20,8):"yon"
940 CALL SOUND(D*3,175,\,440
,\,523,_):: DISPLAY AT(20,12
):"Vir" :: CALL SOUND(D,247,
\,392,\,494,_):: CALL SOUND(
D*2,175,\,349,\,440,_)
950 DISPLAY AT(20,15):"gin";
:: CALL SOUND(D*3,131,\,330,
\,392,_):: DISPLAY AT(20,19)
:"Moth" :: CALL SOUND(D,131,
\,349,\,440,_):: DISPLAY AT(
20,23):"er"
960 CALL SOUND(D*2,131,\,330
,\,392,_):: DISPLAY AT(20,26
):"and" :: CALL SOUND(D*4,13
1,\,262,\,330,_):: DISPLAY A
T(21,@):"Child,"
970 CALL SOUND(D*2,22000,30)
:: CALL SOUND(D*4,175,\,349,
\,440,_):: DISPLAY AT(21,8):
"Ho" :: CALL SOUND(D*2,175,\
,349,\,440,10):: DISPLAY AT(
21,_):"ly"
980 CALL SOUND(D*3,175,\,440
,\,523,_):: DISPLAY AT(21,13
):"In" :: CALL SOUND(D,175,\
,392,\,494,_):: DISPLAY AT(2
1,15):"fant"
990 CALL SOUND(D*2,175,\,349
,\,440,_):: DISPLAY AT(21,20
):"so" :: CALL SOUND(D*3,131
,\,330,\,392,_):: DISPLAY AT
(21,23):"ten"
1000 CALL SOUND(D,131,\,349,
\,440,_):: DISPLAY AT(21,26)
:"der" :: CALL SOUND(D*2,131
,\,330,\,392,_):: DISPLAY AT
(22,@):"and"
1010 CALL SOUND(D*4,131,\,26
2,\,330,_):: DISPLAY AT(22,5
):"mild," :: CALL SOUND(D*2,
30000,30)
1020 CALL SOUND(D*4,196,\,34
9,\,587,_):: DISPLAY AT(22,1
1):"Sleep"
1030 CALL SOUND(D*2,196,\,34
9,\,587,_):: DISPLAY AT(22,1
7):"in" :: CALL SOUND(D*3,19
6,\,349,\,698,_):: DISPLAY A
T(22,20):"heav"
1040 CALL SOUND(D,196,\,349,
\,587,_):: DISPLAY AT(22,24)
:"en" :: CALL SOUND(D*2,196,
\,349,\,494,_):: DISPLAY AT(
22,26):"ly"
1050 CALL SOUND(D*6,131,\,33
0,\,523,_):: DISPLAY AT(23,@
):"peace," :: CALL SOUND(D*4
,131,\,394,\,659,_):: CALL S
OUND(D*2,30000,30)
1060 CALL SOUND(D*3,131,\,33
0,\,523,_):: DISPLAY AT(23,8
):"Sleep" :: CALL SOUND(D,19
6,\,392,\):: CALL SOUND(D*2,
131,\,262,\,330,_):: DISPLAY
 AT(23,14):"in"
1070 CALL SOUND(D*3,196,\,33
0,\,392,_):: DISPLAY AT(23,1
7):"heav" :: CALL SOUND(D,19
6,\,294,\,349,_):: DISPLAY A
T(23,21):"en"
1080 CALL SOUND(D*2,175,\,24
7,\,294,_):: DISPLAY AT(23,2
3):"ly"
1090 CALL SOUND(D*6,131,\,16
5,\,262,_):: DISPLAY AT(24,@
):"peace." :: CALL SOUND(D*4
,131,\,165,\,262,_)
1100 CALL SOUND(D*5,30000,30
):: GOSUB 2410
1110 CALL SOUND(D*3,131,\,33
0,\,392,_):: DISPLAY AT(18,@
):"Si" :: CALL SOUND(D,262,\
,349,\,440,_)
1120 CALL SOUND(D*2,262,\,33
0,\,392,_):: DISPLAY AT(18,3
):"lent" :: CALL SOUND(D*6,1
96,\,262,\,330,_):: DISPLAY
AT(18,8):"night,"
1130 CALL SOUND(D*3,131,\,33
0,\,392,_):: DISPLAY AT(18,1
5):"Ho" :: CALL SOUND(D,262,
\,349,\,440,_)
1140 CALL SOUND(D*2,262,\,33
0,\,392,_):: DISPLAY AT(18,1
7):"ly" :: CALL SOUND(D*4,19
6,\,262,\,330,_):: DISPLAY A
T(18,20):"night,"
1150 CALL SOUND(D*4,196,\,34
9,\,587,_):: DISPLAY AT(19,@
):"Shep" :: CALL SOUND(D*2,1
96,\,349,\,587,_):: DISPLAY
AT(19,5):"herds"
1160 CALL SOUND(D*6,196,\,34
9,\,494,_):: DISPLAY AT(19,1
1):"quake" :: CALL SOUND(D*4
,131,\,330,\,523,_):: DISPLA
Y AT(19,17):"at"
1170 CALL SOUND(D*2,131,\,33
0,\,523,_):: DISPLAY AT(19,2
0):"the" :: CALL SOUND(D*6,1
31,\,330,\,392,_):: DISPLAY
AT(20,@):"sight."
1180 CALL SOUND(D*4,175,\,34
9,\,440,_):: DISPLAY AT(20,8
):"Glo" :: CALL SOUND(D*2,17
5,\,349,\,440,_):: DISPLAY A
T(20,11):"ries"
1190 CALL SOUND(D*3,175,\,44
0,\,523,_):: DISPLAY AT(20,1
6):"stream" :: CALL SOUND(D,
247,\,392,\,494,_):: CALL SO
UND(D*2,175,\,349,\,440,_)
1200 DISPLAY AT(20,23):"from
";:: CALL SOUND(D*3,131,\,33
0,\,392,_):: DISPLAY AT(21,@
):"heav" :: CALL SOUND(D,131
,\,349,\,440,_):: DISPLAY AT
(21,5):"en"
1210 CALL SOUND(D*2,131,\,33
0,\,392,_):: DISPLAY AT(21,8
):"a" :: CALL SOUND(D*4,131,
\,262,\,330,_):: DISPLAY AT(
21,9):"far,"
1220 CALL SOUND(D*2,22000,30
):: CALL SOUND(D*4,175,\,349
,\,440,_):: DISPLAY AT(21,14
):"Heav'n" :: CALL SOUND(D*2
,175,\,349,\,440,_):: DISPLA
Y AT(21,20):"ly"
1230 CALL SOUND(D*3,175,\,44
0,\,523,_):: DISPLAY AT(21,2
3):"hosts" :: CALL SOUND(D,1
75,\,392,\,494,_)
1240 CALL SOUND(D*2,175,\,34
9,\,440,_):: DISPLAY AT(22,@
):"sing" :: CALL SOUND(D*3,1
31,\,330,\,392,_):: DISPLAY
AT(22,6):"al"
1250 CALL SOUND(D,131,\,349,
\,440,_):: DISPLAY AT(22,8):
"le" :: CALL SOUND(D*2,131,\
,330,\,392,_):: DISPLAY AT(2
2,10):"lu"
1260 CALL SOUND(D*4,131,\,26
2,\,330,_):: DISPLAY AT(22,1
2):"ia;" :: CALL SOUND(D*2,3
0000,30)
1270 CALL SOUND(D*4,196,\,34
9,\,587,_):: DISPLAY AT(22,1
6):"Christ"
1280 CALL SOUND(D*2,196,\,34
9,\,587,_):: DISPLAY AT(22,2
3):"the" :: CALL SOUND(D*3,1
96,\,349,\,698,_):: DISPLAY
AT(23,@):"Sa"
1290 CALL SOUND(D,196,\,349,
\,587,_):: DISPLAY AT(23,3):
"vior" :: CALL SOUND(D*2,196
,\,349,\,494,_):: DISPLAY AT
(23,8):"is"
1300 CALL SOUND(D*6,131,\,33
0,\,523,_):: DISPLAY AT(23,1
1):"born," :: CALL SOUND(D*4
,131,\,394,\,659,_):: CALL S
OUND(D*2,30000,30)
1310 CALL SOUND(D*3,131,\,33
0,\,523,_):: DISPLAY AT(23,1
7):"Christ" :: CALL SOUND(D,
196,\,392,\):: CALL SOUND(D*
2,131,\,262,\,330,_):: DISPL
AY AT(23,24):"the"
1320 CALL SOUND(D*3,196,\,33
0,\,392,_):: DISPLAY AT(24,@
):"Sa" :: CALL SOUND(D,196,\
,294,\,349,_):: DISPLAY AT(2
4,3):"vior"
1330 CALL SOUND(D*2,175,\,24
7,\,294,_):: DISPLAY AT(24,8
):"is"
1340 CALL SOUND(D*6,131,\,16
5,\,262,_):: DISPLAY AT(24,1
1):"born." :: CALL SOUND(D*4
,131,\,165,\,262,_)
1350 CALL SOUND(D*5,3000,30)
:: GOSUB 2410
1360 CALL SOUND(D*3,131,\,33
0,\,392,_):: DISPLAY AT(18,@
):"Si" :: CALL SOUND(D,262,\
,349,\,440,_)
1370 CALL SOUND(D*2,262,\,33
0,\,392,_):: DISPLAY AT(18,3
):"lent" :: CALL SOUND(D*6,1
96,\,262,\,330,_):: DISPLAY
AT(18,8):"night,"
1380 CALL SOUND(D*3,131,\,33
0,\,392,_):: DISPLAY AT(18,1
5):"Ho" :: CALL SOUND(D,262,
\,349,\,440,_)
1390 CALL SOUND(D*2,262,\,33
0,\,392,_):: DISPLAY AT(18,1
7):"ly" :: CALL SOUND(D*4,19
6,\,262,\,330,_):: DISPLAY A
T(18,20):"night,"
1400 CALL SOUND(D*4,196,\,34
9,\,587,_):: DISPLAY AT(19,@
):"Son" :: CALL SOUND(D*2,19
6,\,349,\,587,_):: DISPLAY A
T(19,5):"of"
1410 CALL SOUND(D*6,196,\,34
9,\,494,_):: DISPLAY AT(19,8
):"God," :: CALL SOUND(D*4,1
31,\,330,\,523,_):: DISPLAY
AT(19,13):"love’s"
1420 CALL SOUND(D*2,131,\,33
0,\,523,_):: DISPLAY AT(19,2
0):"pure" :: CALL SOUND(D*6,
131,\,330,\,392,_):: DISPLAY
 AT(20,@):"light"
1430 CALL SOUND(D*4,175,\,34
9,\,440,_):: DISPLAY AT(20,7
):"Ra" :: CALL SOUND(D*2,175
,\,349,\,440,_):: DISPLAY AT
(20,9):"diant"
1440 CALL SOUND(D*3,175,\,44
0,\,523,_):: DISPLAY AT(20,1
5):"beams" :: CALL SOUND(D,2
47,\,392,\,494,_):: CALL SOU
ND(D*2,175,\,349,\,440,_)
1450 DISPLAY AT(20,21):"from
";:: CALL SOUND(D*3,131,\,33
0,\,392,_):: DISPLAY AT(20,2
6):"Thy" :: CALL SOUND(D,131
,\,349,\,440,_):: DISPLAY AT
(21,@):"ho"
1460 CALL SOUND(D*2,131,\,33
0,\,392,_):: DISPLAY AT(21,3
):"ly" :: CALL SOUND(D*4,131
,\,262,\,330,_):: DISPLAY AT
(21,6):"face,"
1470 CALL SOUND(D*2,22000,30
):: CALL SOUND(D*4,175,\,349
,\,440,_):: DISPLAY AT(21,12
):"With" :: CALL SOUND(D*2,1
75,\,349,\,440,_):: DISPLAY
AT(21,17):"the"
1480 CALL SOUND(D*3,175,\,44
0,\,523,_):: DISPLAY AT(21,2
1):"dawn" :: CALL SOUND(D,17
5,\,392,\,494,_):: DISPLAY A
T(21,26):"of"
1490 CALL SOUND(D*2,175,\,34
9,\,440,_):: DISPLAY AT(22,@
):"re" :: CALL SOUND(D*3,131
,\,330,\,392,_):: DISPLAY AT
(22,3):"deem"
1500 CALL SOUND(D,131,\,349,
\;440,_):: CALL SOUND(D*2,13
1,\,330,\,392,_):: DISPLAY A
T(22,7):"ing"
1510 CALL SOUND(D*4,131,\,26
2,\,330,_):: DISPLAY AT(22,1
1):"grace," :: CALL SOUND(D*
2,30000,30)
1520 CALL SOUND(D*4,196,\,34
9,\,587,_):: DISPLAY AT(22,1
8):"Je"
1530 CALL SOUND(D*2,196,\,34
9,\,587,_):: DISPLAY AT(22,2
0):"sus," :: CALL SOUND(D*3,
196,\,349,\,698,_):: DISPLAY
 AT(23,@):"Lord,"
1540 CALL SOUND(D,196,\,349,
\,587,_):: DISPLAY AT(23,7):
"at" :: CALL SOUND(D*2,196,\
,349,\,494,_):: DISPLAY AT(2
3,10):"Thy"
1550 CALL SOUND(D*6,131,\,33
0,\,523,_):: DISPLAY AT(23,1
4):"birth," :: CALL SOUND(D*
4,131,\,394,\,659,_):: CALL
SOUND(D*2,30000,30)
1560 CALL SOUND(D*3,131,\,33
0,\,523,_):: DISPLAY AT(23,2
1):"Je" :: CALL SOUND(D,196,
\,392,\):: CALL SOUND(D*2,13
1,\,262,\,330,_):: DISPLAY A
T(23,23):"sus,"
1570 CALL SOUND(D*3,196,\,33
0,\,392,_):: DISPLAY AT(24,@
):"Lord," :: CALL SOUND(D,19
6,\,294,\,349,_):: DISPLAY A
T(24,7):"at"
1580 CALL SOUND(D*2,175,\,24
7,\,294,_):: DISPLAY AT(24,1
0):"Thy"
1590 CALL SOUND(D*6,131,\,16
5,\,262,_):: DISPLAY AT(24,1
4):"birth." :: CALL SOUND(D*
4,131,\,165,\,262,_)
1600 CALL SOUND(D*8,30000,30
)
1610 IF B$<>"YES" THEN 2330
1620 !-I Heard the Bells on
Christmas Day
1630 GOSUB 2410 :: R=17 :: M
$="I Heard the Bells" :: GOS
UB 2420 :: R=R+@ :: M$="on C
hristmas Day" :: GOSUB 2420
1640 R=R+@ :: M$="H.W.Longfe
llow   J.B.Calkin" :: GOSUB
 2420 :: GOSUB 2410 :: D=150
1650 CALL SOUND(D*4,165,\,19
6,\,330,_):: DISPLAY AT(18,@
):"I" :: CALL SOUND(D*6,165,
\,330,\,392,_):: DISPLAY AT(
18,3):"heard"
1660 CALL SOUND(D*2,247,\,33
0,\,370,_):: DISPLAY AT(18,9
```

```
):"the" :: CALL SOUND(D#4,16
5,\,330,\,392,_):: DISPLAY A
T(18,13):"bells"
1670 CALL SOUND(D#4,165,\,27
7,\,392,_):: DISPLAY AT(18,1
9):"on" :: CALL SOUND(D#6,17
5,\,294,\,440,_):: DISPLAY A
T(18,22):"Christ-"
1680 CALL SOUND(D#2,247,\,29
4,\,392,_):: DISPLAY AT(19,@
):"mas" :: CALL SOUND(D#4,17
5,\,294,\,440,_):: DISPLAY A
T(19,5):"day"
1690 CALL SOUND(D#4,185,\,33
0,\,440,_):: DISPLAY AT(19,9
):"Their" :: CALL SOUND(D#4,
196,\,330,\,494,_):: DISPLAY
 AT(19,15):"old"
1700 CALL SOUND(D#4,247,\,39
2,\,659,_):: DISPLAY AT(19,1
9):"fa" :: CALL SOUND(D#4,16
5,\,392,\,587,_):: DISPLAY A
T(19,21):"mil"
1710 CALL SOUND(D#4,196,\,33
0,\,523,_):: DISPLAY AT(19,2
4):"iar" :: CALL SOUND(D#6,2
20,\,349,\,523,_):: DISPLAY
AT(20,@):"ca"
1720 CALL SOUND(D#2,220,\,34
9,\,494,_):: DISPLAY AT(20,3
):"rols" :: CALL SOUND(D#4,2
20,\,349,\,494,_):: DISPLAY
AT(20,8):"play,"
1730 CALL SOUND(D#4,196,\,33
0,\,494,_):: DISPLAY AT(20,1
4):"And" :: CALL SOUND(D#6,2
20,\,330,\,494,_):: DISPLAY
AT(20,18):"wild"
1740 CALL SOUND(D#2,220,\,33
0,\,440,_):: DISPLAY AT(20,2
3):"and" :: CALL SOUND(D#4,2
47,\,330,\,392,_):: DISPLAY
AT(21,@):"sweet"
1750 CALL SOUND(D#4,165,\,24
7,\,440,_):: DISPLAY AT(21,7
):"the" :: CALL SOUND(D#6,14
7,\,247,\,392,_):: DISPLAY A
T(21,11):"words"
1760 CALL SOUND(D#2,175,\,24
7,\,349,_):: DISPLAY AT(21,1
7):"re" :: CALL SOUND(D#4,13
1,\,262,\,330,_):: DISPLAY A
T(21,19):"peat"
1770 CALL SOUND(D#4,131,\,22
0,\,349,_):: DISPLAY AT(21,2
4):"Of" :: CALL SOUND(D#4,12
3,\,330,\,392,_):: DISPLAY A
T(22,@):"peace"
1780 CALL SOUND(D#4,262,\,33
0,\,440,_):: DISPLAY AT(22,7
):"on" :: CALL SOUND(D#4,110
,\,330,\,494,_):: DISPLAY AT
(22,10):"earth,"
1790 CALL SOUND(D#4,110,\,33
0,\,523,_):: DISPLAY AT(22,1
7):"good" :: CALL SOUND(D#o,
123,\,175,\,294,_):: DISPLAY
 AT(22,22):"will"
1800 CALL SOUND(D#2,123,\,29
4,\,349,_):: DISPLAY AT(23,@
):"to" :: CALL SOUND(D#4,165
,\,196,\,330,_):: DISPLAY AT
(23,4):"men."
1810 CALL SOUND(D#4,30000,30
):: GOSUB 2410
1820 CALL SOUND(D#4,165,\,19
6,\,330,_):: DISPLAY AT(18,@
):"I" :: CALL SOUND(D#6,165,
\,330,\,392,_):: DISPLAY AT(
18,3):"thought"
1830 CALL SOUND(D#2,247,\,33
0,\,370,_):: DISPLAY AT(18,1
1):"how," :: CALL SOUND(D#4,
165,\,330,\,392,_):: DISPLAY
 AT(18,16):"as"
1840 CALL SOUND(D#4,165,\,27
7,\,392,_):: DISPLAY AT(18,1
9):"the" :: CALL SOUND(D#6,1
75,\,294,\,440,_):: DISPLAY
AT(18,23):"day"
1850 CALL SOUND(D#2,247,\,29
4,\,392,_):: DISPLAY AT(19,@
):"had" :: CALL SOUND(D#4,17
5,\,294,\,440,_):: DISPLAY A
T(19,5):"come,"
1860 CALL SOUND(D#4,185,\,33
0,\,440,_):: DISPLAY AT(19,1
1):"The" :: CALL SOUND(D#4,1
96,\,330,\,494,_):: DISPLAY
AT(19,15):"bel"
1870 CALL SOUND(D#4,247,\,39
2,\,659,_):: DISPLAY AT(19,1
8):"fries" :: CALL SOUND(D#4
,165,\,392,\,587,_):: DISPLA
Y AT(19,24):"of"
1880 CALL SOUND(D#4,196,\,33
0,\,523,_):: DISPLAY AT(20,@
):"all" :: CALL SOUND(D#6,22
0,\,349,\,523,_):: DISPLAY A
T(20,5):"Christ"
1890 CALL SOUND(D#2,220,\,34
9,\,494,_):: DISPLAY AT(20,1
1):"en" :: CALL SOUND(D#4,22
0,\,349,\,494,_):: DISPLAY A
T(20,13):"dom"
1900 CALL SOUND(D#4,196,\,33
0,\,494,_):: DISPLAY AT(20,1
7):"Had" :: CALL SOUND(D#6,2
20,\,330,\,494,_):: DISPLAY
AT(20,21):"roll'd"
1910 CALL SOUND(D#2,220,\,33
0,\,440,_):: DISPLAY AT(21,@
):"a" :: CALL SOUND(D#4,247,
\,330,\,392,_):: DISPLAY AT(
21,2):"long"
1920 CALL SOUND(D#4,165,\,24
7,\,440,_):: DISPLAY AT(21,7
):"th'un" :: CALL SOUND(D#6,
147,\,247,\,392,_):: DISPLAY
 AT(21,12):"bro"
1930 CALL SOUND(D#2,175,\,24
7,\,349,_):: DISPLAY AT(21,1
5):"ken" :: CALL SOUND(D#4,1
31,\,262,\,330,_):: DISPLAY
AT(21,19):"song"
1940 CALL SOUND(D#4,131,\,22
0,\,349,_):: DISPLAY AT(21,2
4):"Of" :: CALL SOUND(D#4,12
3,\,330,\,392,_):: DISPLAY A
T(22,@):"peace"
1950 CALL SOUND(D#4,262,\,33
0,\,440,_):: DISPLAY AT(22,7
):"on" :: CALL SOUND(D#4,110
,\,330,\,494,_):: DISPLAY AT
(22,10):"earth,"
1960 CALL SOUND(D#4,110,\,33
0,\,523,_):: DISPLAY AT(22,1
7):"good" :: CALL SOUND(D#6,
123,\,175,\,294,_):: DISPLAY
 AT(22,22):"will"
1970 CALL SOUND(D#2,123,\,29
4,\,349,_):: DISPLAY AT(23,@
):"to" :: CALL SOUND(D#4,165
,\,196,\,330,_):: DISPLAY AT
(23,4):"men."
1980 CALL SOUND(D#4,30000,30
):: GOSUB 2410
1990 CALL SOUND(D#4,165,\,19
6,\,330,_):: DISPLAY AT(18,@
):"And" :: CALL SOUND(D#6,16
5,\,330,\,392,_):: DISPLAY A
T(18,5):"in"
2000 CALL SOUND(D#2,247,\,33
0,\,370,_):: DISPLAY AT(18,8
):"de" :: CALL SOUND(D#4,165
,\,330,\,392,_):: DISPLAY AT
(18,10):"spair"
2010 CALL SOUND(D#4,165,\,27
7,\,392,_):: DISPLAY AT(18,1
6):"I" :: CALL SOUND(D#6,175
,\,294,\,440,_):: DISPLAY AT
(18,18):"bow'd"
2020 CALL SOUND(D#2,247,\,29
4,\,392,_):: DISPLAY AT(18,2
4):"my" :: CALL SOUND(D#4,17
5,\,294,\,440,_):: DISPLAY A
T(19,@):"head;"
2030 CALL SOUND(D#4,185,\,33
0,\,440,_):: DISPLAY AT(19,7
):"""There" :: CALL SOUND(D#
4,[illegible]
[illegible]
2040 CALL SOUND(D#4,247,\,39
2,\,659,_):: DISPLAY AT(19,1
7):"no" :: CALL SOUND(D#4,16
5,\,392,\,587,_):: DISPLAY A
T(19,20):"peace"
2050 CALL SOUND(D#4,196,\,33
0,\,523,_):: DISPLAY AT(19,2
6):"on" :: CALL SOUND(D#6,22
0,\,349,\,523,_):: DISPLAY A
T(20,@):"earth,"""
2060 CALL SOUND(D#2,220,\,34
9,\,494,_):: DISPLAY AT(20,9
):"I" :: CALL SOUND(D#4,220,
\,349,\,494,_):: DISPLAY AT(
20,11):"said,"
2070 CALL SOUND(D#4,196,\,33
0,\,494,_):: DISPLAY AT(20,1
7):"""For" :: CALL SOUND(D#6
,220,\,330,\,494,_):: DISPLA
Y AT(20,22):"hate"
2080 CALL SOUND(D#2,220,\,33
0,\,440,_):: DISPLAY AT(20,2
7):"is" :: CALL SOUND(D#4,24
7,\,330,\,392,_):: DISPLAY A
T(21,@):"strong,"
2090 CALL SOUND(D#4,165,\,24
7,\,440,_):: DISPLAY T(21,9)
:"and" :: CALL SOUND(D#6,147
,\,247,\,392,_):: DISPLAY AT
(21,13):"mocks"
2100 CALL SOUND(D#2,175,\,24
7,\,349,_):: DISPLAY AT(21,1
9):"the" :: CALL SOUND(D#4,1
31,\,262,\,330,_):: DISPLAY
AT(21,23):"song"
2110 CALL SOUND(D#4,131,\,22
0,\,349,_):: DISPLAY AT(22,@
):"Of" :: CALL SOUND(D#4,123
,\,330,\,392,_):: DISPLAY AT
(22,4):"peace"
2120 CALL SOUND(D#4,262,\,33
0,\,440,_):: DISPLAY AT(22,1
0):"on" :: CALL SOUND(D#4,11
0,\,330,\,494,_):: DISPLAY A
T(22,13):"earth,"
2130 CALL SOUND(D#4,110,\,33
0,\,523,_):: DISPLAY AT(22,2
0):"good" :: CALL SOUND(D#6,
123,\,175,\,294,_):: DISPLAY
 AT(22,25):"will"
2140 CALL SOUND(D#2,123,\,29
4,\,349,_):: DISPLAY AT(23,@
):"to" :: CALL SOUND(D#4,165
,\,196,\,330,_):: DISPLAY AT
(23,4):"men."""
2150 CALL SOUND(D#4,30000,30
):: GOSUB 2410
2160 CALL SOUND(D#4,165,\,19
[illegible]
65,\,330,\,392,_):: DISPLAY
AT(18,6):"pealed"
2170 CALL SOUND(D#2,247,\,33
0,\,370,_):: DISPLAY AT(18,1
3):"the" :: CALL SOUND(D#4,1
65,\,330,\,392,_):: DISPLAY
AT(18,17):"bells"
2180 CALL SOUND(D#4,165,\,27
7,\,392,_):: DISPLAY AT(18,2
3):"more" :: CALL SOUND(D#6,
175,\,294,\,440,_):: DISPLAY
 AT(19,@):"loud"
2190 CALL SOUND(D#2,247,\,29
4,\,392,_):: DISPLAY AT(19,6
):"and" :: CALL SOUND(D#4,17
5,\,294,\,440,_):: DISPLAY A
T(19,10):"deep:"
2200 CALL SOUND(D#4,185,\,33
0,\,440,_):: DISPLAY AT(19,1
6):"""God" :: CALL SOUND(D#4
,196,\,330,\,494,_):: DISPLA
Y AT(19,21):"is"
2210 CALL SOUND(D#4,247,\,39
2,\,659,_):: DISPLAY AT(19,2
4):"not" :: CALL SOUND(D#4,1
65,\,392,\,587,_):: DISPLAY
AT(20,@):"desd,"
2220 CALL SOUND(D#4,196,\,33
0,\,523,_):: DISPLAY AT(20,7
):"nor" :: CALL SOUND(D#6,22
0,\,349,\,523,_):: DISPLAY A
T(20,11):"doth"
2230 CALL SOUND(D#2,220,\,34
9,\,494,_):: DISPLAY AT(20,1
6):"he" :: CALL SOUND(D#4,22
0,\,349,\,494,_):: DISPLAY A
T(20,19):"sleep;"
2240 CALL SOUND(D#4,196,\,33
0,\,494,_):: DISPLAY AT(20,2
6):"The" :: CALL SOUND(D#6,2
20,\,330,\,494M_):: DISPLAY
AT(21,@):"wrong"
2250 CALL SOUND(D#2,220,\,33
0,\,440,_):: DISPLAY AT(21,7
):"shall" :: CALL SOUND(D#4,
247,\,330,\,392,_):: DISPLAY
 AT(21,13):"fail,"
2260 CALL SOUND(D#4,165,\,24
7,\,440,_):: DISPLAY AT(21,1
9):"the" :: CALL SOUND(D#6,1
47,\,247,\,392,_):: DISPLAY
AT(21,23):"right"
2270 CALL SOUND(D#2,175,\,24
7,\,349,_):: DISPLAY AT(22,@
):"pre" :: CALL SOUND(D#4,13
1,\,262,\,330,_):: DISPLAY A
T(22,4):"vail,"
2280 CALL SOUND(D#4,131,\,22
0,\,349,_):: DISPLAY AT(22,1
0):"With" :: CALL SOUND(D#4,
123,\,330,\,392,_):: DISPLAY
 AT(22,15):"peace"
2290 CALL SOUND(D#4,262,\,33
0,\,440,_):: DISPLAY AT(22,2
1):"on" :: CALL SOUND(D#4,11
0,\,330,\,494,_):: DISPLAY A
T(23,@):"earth,"
2300 CALL SOUND(D#4,110,\,33
0,\,523,_):: DISPLAY AT(23,8
,):"good" :: CALL´SOUND(D#6,
123,\,175,\,294,_):: DISPLAY
 AT(23,13):"will"
2310 CALL SOUND(D#2,123,\,29
4,\,349,_):: DISPLAY AT(23,1
8):"to" :: CALL SOUND(D#4,16
5,\,196,\,330,_):: DISPLAY A
T(23,21):"men."
2320 CALL SOUND(D#4,30000,30
)
2330 FOR D=@ TO 10
2340 DISPLAY AT(17,9):"AGAIN
 [Y-N]"
2350 CALL KEY(0,K,ST):: IF S
T<>0 THEN 2390
2360 CALL HCHAR(17,9,32,14)
2370 NEXT D
2380 IF ST=0 THEN 2330
2390 YN$=CHR$(K):: IF YN$="Y
" THEN BOTH$="" :: GOTO 270
2400 IF YN$<>"N" THEN 2350 E
LSE CALL CLEAR :: M$="""HAPP
Y CHRISTMAS to all" :: GOSUB
 2420 :: R=R+4 :: M$="and to
 all a GOOD NIGHT""" :: GOSU
B 2420 :: FOR D=@ TO 500 ::
NEXT D :: END
2410 CALL HCHAR(18,@,32,224)
:: RETURN
2420 A=LEN(M$):: Z=(32-A)/2
:: FOR B=@ TO A :: C$=SEG$(M
$,B,@):: C=ASC(C$):: IF C=32
 THEN 2440
2430 CALL SOUND(@,-2,0)
2440 CALL HCHAR(R,B+Z,C)
2450 NEXT B :: FOR D=@ TO 30
0 :: NEXT D :: RETURN
2460 END
```



(Thanks to Lou Amadio and the Tshug UG of New South Wales)

Notes:

1) The FCTN[R] switch will generate a "[" character when the keyboard is in normal mode.

2) Using the above technique, it should be possible to emulate the cursor control keys. The switches could be mounted on the module port bay next to the <ENTER> key. However, in this instance, the number of relays required is too cumbersome, so a more elegant (electronic) solution is required.

Special Function Key Emulation

by Lou Amadio

Encouraged by Derek Wilkinson's effort at interfacing an IBM keyboard to the TI99/4A and the special dual contact keys that he had to modify, I experimented with a way of providing CTRL[U] and FCTN[R] using just one keystroke for each. As you may recall, these functions are used in TI-Writer for the following.

CTRL[U] changes the cursor to Special Character Mode. This mode allows the generation of ASCII characters below 32 - that is control characters.

FCTN[R] generates the Escape character (>1b in Special Character Mode). Escape, in conjunction with other keyboard characters, is used to invoke the special features of your printer such as double strike or emphasized print, etc.

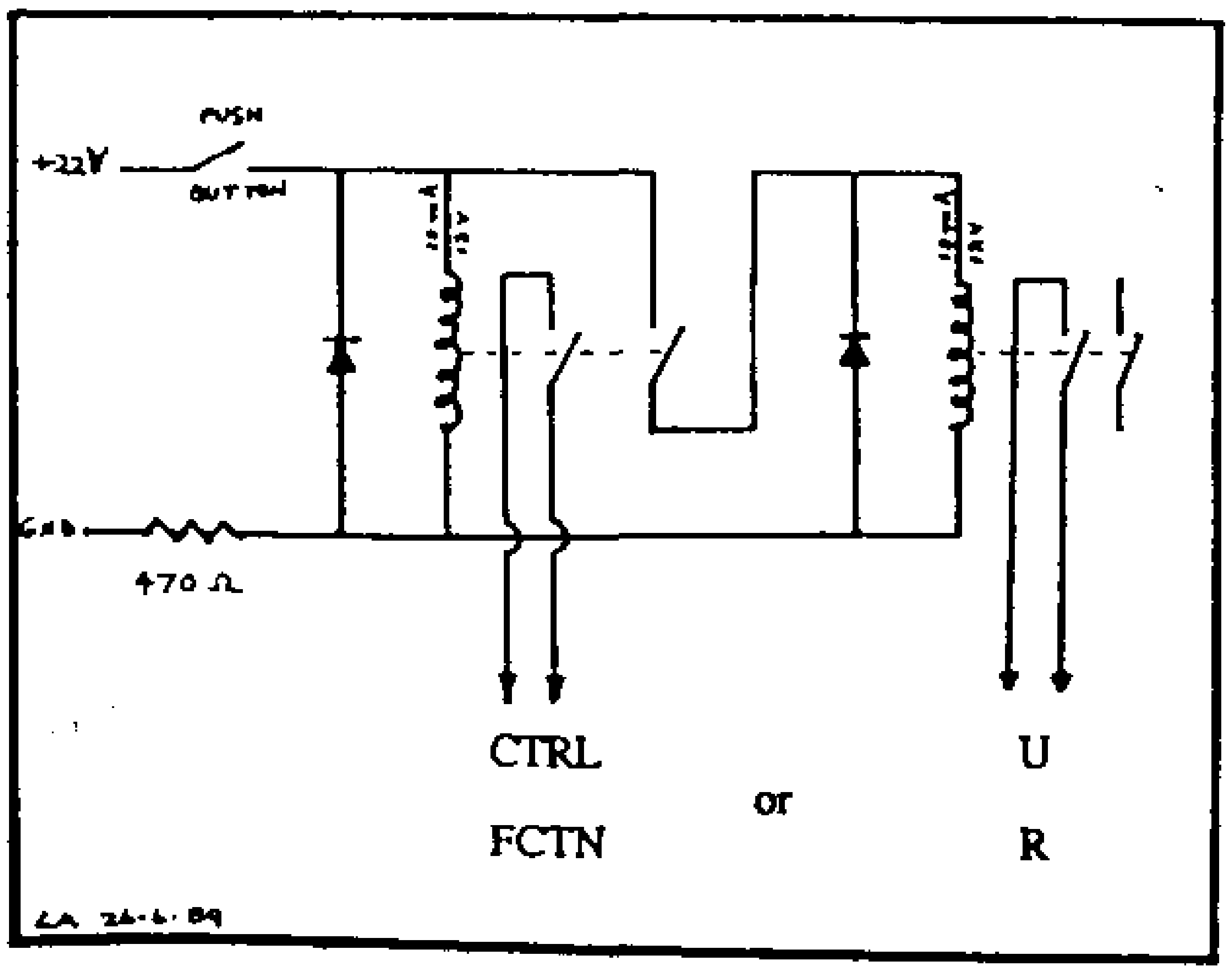
I initially tried to emulate a dual key action by wiring the normally open contacts of a double pole relay across the appropriate key switches on the back of the keyboard. A small momentary action push button switch, in series with a resistor, was used to power the relay from the 22 volt console supply.

Operation was, however, unreliable, as I found that the computer would sometimes generate the wrong character or just intermittently lock up.

In order to introduce a time delay into the second "keystroke", I decided to use the second set of contacts on the relay to power another relay. The normally open contacts of this second relay were then wired across the second key switch. This process proved to be very reliable, provided that a diode and capacitor were soldered across the relay coils to prevent the back emf from getting into the computer's power supply.

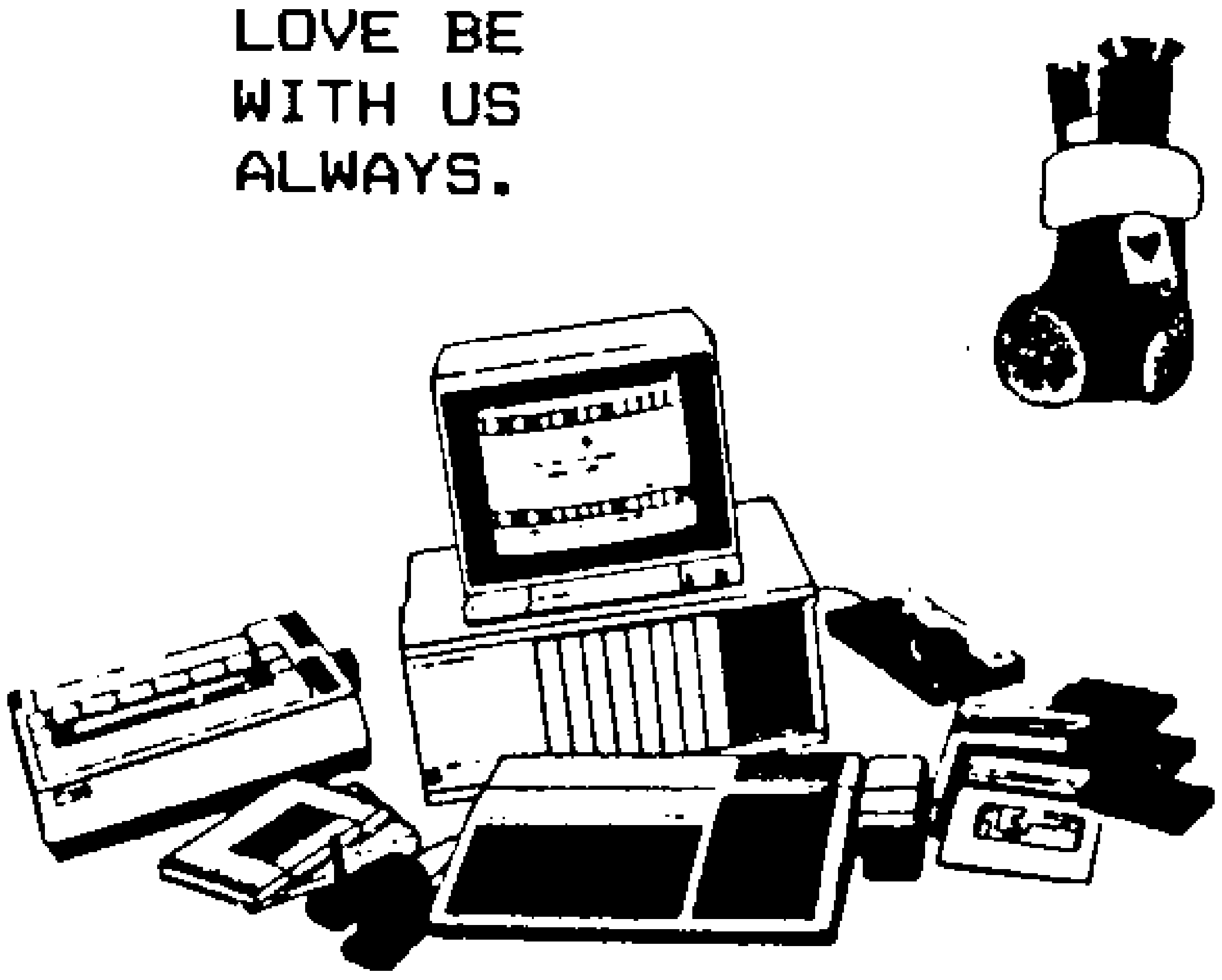
I mounted the 4 miniature relays required for the two functions on a small piece of Veroboard. The board was located above the console power supply and powered via a suitable resistor. The relays that I used were rated at 10 mA at 12 volts. Thus a 470 ohm resistor was used between the 22 volt supply and the push button switch.

The miniature push-button switches were mounted on the upper console shell. The CTRL[U] switch was located to the left of the "A" key and the FCTN[R] switch was located to the right of the "=" key. I used a red push button for CTRL[U] and a black push button for the FCTN[R] as these closely resemble the coloured spots on these keys.



```
MOLOOP MOV *RO+,*R1+  Move code 2 bytes at a time
                         (These 3 lines could be
       DEC R2      Count down (replaced by BL @LOOP1
       JNE MOLOOP Finished yet? (in the following source
       B @>3910          Branch to code
MOVST          Start of code being relocated
*    .  .  .   End of code being relocated
MOVFIN
*________________________________________
* LOADER SOURCE CODE.
* S=MS O=MO
ENDVDP EQU >8370 pointer to end of vram buffer
VMBR   EQU >202C vram multiple byte read
VMBW   EQU >2024 Vram multiple byte write
VWTR   EQU >2030 VDP write to register
       AORG >FEFO i.e.  >FFE8 minus the length of this
*routine = >FEFO with a bit of space to spare for
*extra code.  The following entry points are just like
*the X/B DEF table.  The first 6 bytes are the link
*name and the next 2 bytes are the entry point
*address.  The CALL LINK(8196,254,240) etc in the X/B
*program places the address of the name to link to in
*the normal pointer to the start of the DEF table.
*When the CALL LINK("SEWER2") etc occurs, it is found
*at the first test.
       TEXT 'SEWER2' link name (at >FEFO - 254,240)
       DATA PART2
       TEXT 'SEWER3' link name (at >FEF8 - 254,248)
       DATA PART3
       TEXT 'SEWER4' link name (at >FFOO - 255,0)
       DATA PART4
LOOP1 MOV *RO+,*R1+ Subroutine to transfer code
       DECT R2
       JNE LOOP1
       RT
PART2 MOV R11,@SAVRTN Save return address
       LWPI >2600     Load my workspace
       LI RO,>DEF6    Start of implanted code in 1XB
       LI R1,>A000    destination address in CPU RAM
       LI R2,>1FFA    length of code to move
       BL @LOOP1      Do the move
       MOV @ENDVDP,RO Store end of VDP address in RO
       AI RO,->1FF9   1 less than length of code find
                         start of 2XB code in VDP
       LI R1,>BFFA    destination ram address
       LI R2,>1FFA    length of code
       BLWP @VMBR     transfer to ram
       JMP RETURN
PART3 MOV R11,@SAVRTN
       LWPI >2600     my workspace
       MOV @ENDVDP,RO Store end of VDP address in RO
       AI RO,->19F9   1 less than length of code find
                         start of 3XB code in VDP
       LI R1,>DFF4    destination ram address
       LI R2,>19FA    length of code
       BLWP @VMBR     transfer to ram
RETURN CLR RO         Standard return to Extended BASIC
       MOVB RO,@>837C
       LWPI >83E0
       MOV @SAVRTN,R11
       B *R11
PART4 LWPI >2600      my workspace
       MOV @ENDVDP,RO Store end of VDP address in RO
       AI RO,->0DF9   1 less than length of code find
                         start of 4XB code in VDP
       LI R1,>2800    destination ram address
       LI R2,>0DFA    length of code
       BLWP @VMBR     transfer to ram
       LI RO,>03F0    from
       LI R1,>3600    to BUFFER
       LI R2,>0310    how many
       BLWP @VMBR     read character definitions
       LI RO,>08F0    back to
       BLWP @VMBW     relocate to E/A position
       LI RO,>030E
       BLWP @VWTR     E/A default color table
       LI RO,>0401
       BLWP @VWTR     E/A default pattern table
       LI RO,>07F5
       BLWP @VWTR     E/A default screen color
       B @>A000       execute PROGRAM
SAVRTN BSS 2          When assembled this MUST be at or
                         below >FFE6
       END
```

*
***
CONNI
SENDS T
O YOU AND
YOURS A MER
RY CHRISTMAS
AND A HAPPY NEW
YEAR. MAY YOU BE
RICHLY BLESSED WITH
HEALTH AND HAPPINESS
IN THE YEARS AHEAD. BE
EVER THOUGHTFUL TO YOUR
FELLOWMAN IN WORD AND DEED.
JOY
PEACE
AND HIS
LOVE BE
WITH US
ALWAYS.



DOCUMENTATION FOR CASSETTE INDEX UTILITY VERSION 1.0 (XB)
CREATED BY HARRY BRASHEAR OF THE WESTERN NY 99'ERS
MODIFIED (V.2.0) BY MICKEY SCHMITT OF THE WEST PENN 99'ERS
===

PROGRAM NAME: CS1*INDEX (SEE LIBRARIAN)
PROGRAM DOCS: CS1*DOCS (DOCUMENTATION)
PROGRAM FILES: MOCKFILE1 AND MOCKFILE2 (BASIC SET-UPS)

THIS EXTENDED BASIC PROGRAM WILL ALLOW YOU TO PRINT OUT A NEAT LITTLE INSERT FOR YOUR CASSETTE TAPES, JUST LIKE THE ONES THAT COME WITH THEM.

ALL PRINTER CODES ARE EPSON COMPATIBLE. YOU MAY WISH TO MODIFY PROGRAM LINE 570 TO OPEN YOUR PARTICULAR PRINTER. CURRENTLY THIS PROGRAM OPENS THE FOLLOWING PRINTER DEVICE: (RS232/2.DA=8.BA=9600).

WHEN YOU FIRST BOOT UP THE PROGRAM, YOU'LL BE ASKED IF YOU WANT TO USE FILES OR INPUT THE INFORMATION DIRECTLY. (MORE ON THIS LATER). YOU WILL THEN BE ASKED FOR YOUR FULL NAME, FOR GOOD REASON. THAT LITTLE FOLD OVER THE OTHER SIDE OF YOUR TAPE WILL BE PUT TO GOOD USE BY STATING WHO THE TAPE BELONGS TO. (THE DEFAULT SPACE LIMIT IS 28 CHARACTERS).

NEXT YOU WILL BE ASKED IF YOU WANT ONE OR BOTH SIDES OF THE TAPE INDEXED. (THE DEFAULT IS 2 FOR BOTH SIDES).

YOU WILL THEN BE ASKED FOR THE CASSETTE TITLE (THE DEFAULT SPACE LIMIT IS 25 CHARACTERS), AND ANY SPECIAL INFORMATION THAT YOU MAY WISH TO RECORD. (AGAIN, THE DEFAULT SPACE LIMIT IS 25 CHARACTERS).

FINALLY, YOU WILL BE ABLE TO INPUT YOUR CASSETTE PROGRAMS, USING ANY CODING SYSTEM YOU MAY PREFER. (THE DEFAULT SPACE LIMIT IS 27 CHARACTERS).

SHOULD YOU MAKE A MISTAKE WHILE ENTERING YOUR PROGRAM DATA JUST HIT THE (ENTER KEY) AT A BLANK LINE AND YOU WILL THEN BE PROMPTED BACK TO YOUR PREVIOUS ENTRY.

WHEN YOU HAVE COMPLETED ALL OF YOUR PROGRAM DATA INPUTS, JUST TYPE (Q) AND HIT THE (ENTER KEY). (NOTE: THE (Q) WILL NOT APPEAR ON YOUR PRINTED INSERT).

LAST, YOU WILL BE ASKED IF YOU WOULD LIKE TO FILE OR PRINT YOUR INPUTS. (THIS SPECIAL FEATURE WAS INCLUDED BY HARRY SO THAT PEOPLE WITHOUT PRINTERS COULD FILE THE INFORMATION TILL THEY COULD BORROW A PRINTER OR CON A FRIEND INTO PRINTING THE INSERTS).

NOTE: IF YOU HAVE FILED INFORMATION STORED ON YOUR DISK YOU COULD JUST TYPE (F) AT THE FIRST PROMPT, WHICH WOULD PROMPT YOU FOR THE FILENAME YOU HAVE YOUR DATA STORED UNDER. ONCE YOU HAVE ENTERED YOUR FILENAME, JUST SIT BACK AND WATCH YOUR PRINTER GO INTO ACTION. (PROVIDED YOU DID REMEMBER TO TURN YOUR PRINTER ON).

YOU MAY ALSO MAKE MULTIPLE COPIES BY SIMPLY REPRINTING. THE "BUTTONS" ARE JUST FOR FUN, AND YOU WILL HAVE TO HOLD YOUR KEY PRESSES LONGER BECAUSE OF THEM.

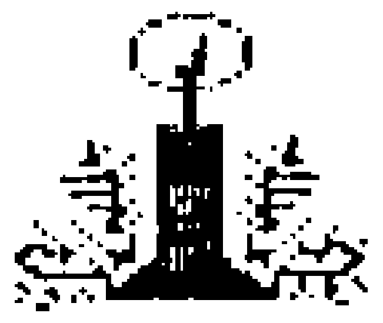
```
----------------------------------------------------------------
!  SIDE 1                          !  SIDE 2                   !
----------------------------------------------------------------
!                                  !                           !
!  123456789012345678901234567     !  123456789012345678901234567
!  2                               !  2                        !
!  3                               !  3                        !
!  4                               !  4                        !
!  5                               !  5                        !
!  6                               !  6                        !
!  7                               !  7                        !
!  8                               !  8                        !
!  9                               !  9                        !
!  0                               !  0                        !
!  1                               !  1                        !
!  2                               !  2                        !
!  3                               !  3                        !
!  4                               !  4                        !
!  523456789012345678901234567     !  523456789012345678901234567
!                                  !                           !
----------------------------------------------------------------
!  SIDE 1:   1234567890123456789012345 1234567890123456789012345 !
!                                                              !
!  SIDE 2:   1234567890123456789012345 1234567890123456789012345 !
----------------------------------------------------------------
!                                                              !
!  THIS TAPE IS FROM THE LIBRARY OF 1234567890123456789012345678 !
!                                                              !
----------------------------------------------------------------
```

MOCKFILE1

```
----------------------------------------------------------------
!  SIDE 1                          !  SIDE 2                   !
----------------------------------------------------------------
!                                  !                           !
!  PROGRAM 001 - COUNTER 000 B     !  PROGRAM 016 - COUNTER 000 B !
!  PROGRAM 002 - COUNTER 010 B     !  PROGRAM 017 - COUNTER 010 B !
!  PROGRAM 003 - COUNTER 020 B     !  PROGRAM 018 - COUNTER 020 B !
!  PROGRAM 004 - COUNTER 030 B     !  PROGRAM 019 - COUNTER 030 B !
!  PROGRAM 005 - COUNTER 040 B     !  PROGRAM 020 - COUNTER 040 B !
!  PROGRAM 006 - COUNTER 050 B     !  PROGRAM 021 - COUNTER 050 B !
!  PROGRAM 007 - COUNTER 060 B     !  PROGRAM 022 - COUNTER 060 B !
!  PROGRAM 008 - COUNTER 070 B     !  PROGRAM 023 - COUNTER 070 B !
!  PROGRAM 009 - COUNTER 080 B     !  PROGRAM 024 - COUNTER 080 B !
!  PROGRAM 010 - COUNTER 090 B     !  PROGRAM 025 - COUNTER 090 B !
!  PROGRAM 011 - COUNTER 100 B     !  PROGRAM 026 - COUNTER 100 B !
!  PROGRAM 012 - COUNTER 110 B     !  PROGRAM 027 - COUNTER 110 B !
!  PROGRAM 013 - COUNTER 120 B     !  PROGRAM 028 - COUNTER 120 B !
!  PROGRAM 014 - COUNTER 130 B     !  PROGRAM 029 - COUNTER 130 B !
!  PROGRAM 015 - COUNTER 140 B     !  PROGRAM 030 - COUNTER 140 B !
!                                  !                           !
----------------------------------------------------------------
!  SIDE 1:   WEST PENN 99'ERS CASSETTE LIBRARY PROGRAMS 001--015 !
!                                                              !
!  SIDE 2:   WEST PENN 99'ERS CASSETTE LIBRARY PROGRAMS 016--030 !
----------------------------------------------------------------
!                                                              !
!  THIS TAPE IS FROM THE LIBRARY OF THE WEST PENN 99'ERS LIBRARY !
!                                                              !
----------------------------------------------------------------
```

MOCKFILE2

```
100 !*************************
110 !                        *
120 !CASSETTE INDEX UTILITY*
130 !          BY            *
140 !    HARRY BRASHEAR      *
150 !    WESTERN NY 99ERS    *
160 !                        *
170 !*************************
180 !                        *
190 !    VERSION 2.0 (XB)    *
200 ! ENHANCED AND MODIFED   *
210 !          BY            *
```



(Thanks to West Penn 99'ers
UG and Mickey Schmitt

```
220 !    MICKEY SCHMITT       *
230 !   WEST PENN 99'ERS      *
240 !     JUNE 13 1989        *
250 !                         *
260 !************************
270 ON WARNING NEXT :: CALL
MAGNIFY(3)
280 GOTO 290 :: OPTION BASE
1 :: A$,A1$,A2$,AN$,F$,FN$,L
$,L1$,N$,PF$,T1$,T2$ :: I,J,
K,S,S1,S2,SS :: CALL CHAR ::
 CALL CLEAR :: CALL COLOR ::
 CALL SCREEN :: CALL SPRITE
:: CALL KEY :: DIM SIDE1$(18
),SIDE2$(18):: !@P-
290 !
300 CALL CLEAR :: CALL SCREE
N(15):: CALL CHAR(132,"3F60C
F9FBFBFBFBFBFBFBFBF9FCF603FF
C06F3F9FDFDFDFDFDFDFDFDF9F30
6FC")
310 CALL SPRITE(#1,132,2,160
,47):: CALL SPRITE(#2,132,2,
160,119):: CALL SPRITE(#3,13
2,2,160,190)
320 CALL COLOR(13,7,15):: CA
LL CHAR(128,"000000FFFF",129
,"0000FF00FF00FF"):: L$=RPT$
(CHR$(128),28):: L1$=RPT$(CH
R$(129),28)
330 DISPLAY AT(24,3):"RECORD
  PLAYBACK   DUPE"
340 DISPLAY AT(1,1):"CASSETT
E TAPE--INDEX UTILITY":L1$:
: :"WOULD YOU LIKE TO PRINT
FROM": :"FILES OR DIRECT INP
UT? F/D D"
350 ACCEPT AT(7,28)SIZE(-1)B
EEP VALIDATE("FD"):F$ :: IF
F$="F" THEN 730 ELSE CALL SP
RITE(#1,132,10,160,47)
360 DISPLAY AT(4,1):"PLEASE
ENTER YOUR FULL NAME:":L$ ::
 ACCEPT AT(7,1)SIZE(28)BEEP:
N$ :: DISPLAY AT(8,1):L$
370 DISPLAY AT(4,1):"ENTER N
UMBER OF TAPE SIDES:2":L$ ::
 ACCEPT AT(4,28)BEEP VALIDAT
E("12")SIZE(-1):SS
380 DISPLAY AT(4,1):"ENTER C
ASSETTE TITLE: SIDE 1" :: AC
CEPT AT(9,1)SIZE(25)BEEP:T1$
 :: DISPLAY AT(4,1):"SPECIAL
 INFORMATION:  SIDE 1"
390 ACCEPT AT(10,1)SIZE(25)B
EEP:A1$
400 IF SS=2 THEN DISPLAY AT(
11,1):L$ ELSE 420
410 DISPLAY AT(4,1):"ENTER C
ASSETTE TITLE: SIDE 2":L$ ::
 ACCEPT AT(12,1)SIZE(2[illegible]
:T2$ :: DISPLAY AT(4,1):"SPE
CIAL INFOMATION:   SIDE 2" :
: ACCEPT AT(13,1)SIZE(25)BEE
P:A2$
420 DISPLAY AT(16,1)BEEP:"AL
L INFORMATION CORRECT? Y/N"
430 CALL KEY(0,K,S):: IF S=0
 THEN 430 ELSE IF K=89 THEN
440 ELSE IF K=78 THEN 270 EL
SE 430
440 A$="SIDE ONE" :: GO SUB
670
450 FOR I=5 TO 19 :: J=I-4 :
: S1=J :: ACCEPT AT(I,2)SIZE
(27)BEEP:SIDE1$(J):: IF SIDE
1$(J)="Q" THEN 480 ELSE IF S
IDE1$(J)="" THEN I=I-2
460 IF I<4 THEN I=5
470 NEXT I
480 IF SS=1 THEN 520 ELSE A$
="SIDE TWO" :: GO SUB 670
490 FOR I=5 TO 19 :: J=I-4 :
: S2=J :: ACCEPT AT(I,2)SIZE
(27)BEEP:SIDE2$(J):: IF SIDE
2$(J)="Q" THEN 520 ELSE IF S
IDE2$(J)="" THEN I=I-2
500 IF I<4 THEN I=5
510 NEXT I
520 DISPLAY AT(24,3)BEEP:" F
ILE     REDO      PRINT" :: CA
LL SPRITE(#1,132,2,160,47)
530 DISPLAY AT(24,10)SIZE(10
):"   REDO" :: CALL KEY(5,K,
S):: CALL SPRITE(#1,132,2,16
0,47,#2,132,2,160,119,#3,132
,2,160,190)
540 IF S=0 THEN 550 ELSE IF
K=70 THEN 690 ELSE IF K=80 T
HEN 560 ELSE IF K=82 THEN 66
0 ELSE 530
550 FOR I=1 TO 80 :: NEXT I
:: CALL SPRITE(#1,132,10,160
,47,#2,132,10,160,119,#3,132
,10,160,190):: GOTO 530
560 CALL SPRITE(#1,132,2,160
,47,#2,132,2,160,119,#3,132,
10,160,190)
570 OPEN #1:"RS232/2.DA=8.BA
=9600" :: PRINT #1:CHR$(27);
CHR$(64);CHR$(27);CHR$(85);C
HR$(1);
580 PRINT #1:CHR$(27);CHR$(4
8);CHR$(15);RPT$("-",70):"!"
;TAB(5);"SIDE 1";TAB(35);"!"
;TAB(39);"SIDE 2";TAB(70);"!
" :: PRINT #1:RPT$("-",70)
590 PRINT #1:"!";TAB(35);"!"
;TAB(70);"!"
600 FOR I=1 TO 17 :: IF SIDE
[illegible]
[illegible]
DE2$(I)=""
620 PRINT #1:"!   ";SIDE1$(I
);TAB(35);"!   ";SIDE2$(I);T
AB(70);"!" :: NEXT I
630 PRINT #1:RPT$("-",70):"!
";"   SIDE 1:     ";T1$;" ";A
1$;TAB(70);"!":"!";TAB(70);"
!":"!";"   SIDE 2:     ";T2$;
" ";A2$;TAB(70);"!":RPT$("-"
,70)
640 PRINT #1:"!";TAB(70);"!"
:"!";"   THIS TAPE IS FROM T
HE LIBRARY OF ";N$;TAB(70);"
!":"!";TAB(70);"!":RPT$("-",
70):: CLOSE #1
650 GOTO 530
660 FOR I=1 TO 15 :: SIDE1$(
I),SIDE2$(I)="" :: NEXT I ::
 T1$,T2$,A1$,A2$="" :: GOTO
270
670 DISPLAY AT(4,1):"" :: DI
SPLAY AT(1,1):"ENTER PROGRAM
 NAMES: Q(QUIT)":L$:TAB(11);
A$
680 FOR I=5 TO 19 :: DISPLAY
 AT(I,1):")" :: NEXT I :: RE
TURN
690 CALL SPRITE(#1,132,10,16
0,47):: DISPLAY AT(1,1):"FIL
E NAME:   DSK1." :: ACCEPT A
T(1,19)BEEP SIZE(10):FN$
700 FN$="DSK1."&FN$ :: OPEN
#2:FN$,SEQUENTIAL,INTERNAL,O
UTPUT,VARIABLE 80
710 PRINT #2:N$ :: PRINT #2:
T1$,A1$ :: PRINT #2:T2$,A2$
720 FOR I=1 TO 15 :: PRINT #
2:SIDE1$(I),SIDE2$(I):: NEXT
 I :: CLOSE #2 :: GOTO 530
730 CALL SPRITE(#2,132,10,16
0,119):: DISPLAY AT(7,1):""
:: DISPLAY AT(4,1):"FILE NAM
E:    DSK1.":L$ :: ACCEPT AT(
4,19)SIZE(10):FN$ :: FN$="DS
K1."&FN$
740 CALL SPRITE(#2,132,10,16
0,119):: OPEN #1:FN$,SEQUENT
IAL,INTERNAL,INPUT ,VARIABLE
 80
750 INPUT #1:N$ :: INPUT #1:
T1$,A1$ :: INPUT #1:T2$,A2$
760 FOR I=1 TO 15 :: IF EOF(
1)THEN 780 ELSE INPUT #1:SID
E1$(I),SIDE2$(I)
770 NEXT I
780 CLOSE #1 :: GOTO 560
790 !@P+
800 END
```

MORE ON 4D GRAPHICS
by Jim Peterson

It seems that my previous article on creating 4-dimensional graphics has created some controversy. Programmers have reported no success in creating these graphics, and some have even questioned my logic.

In fact, these programmers may have succeeded without knowing it. I regret to report that a program to create 4-D graphics will result in nothing more than a blank screen because I have now learned that 4-D objects are invisible!

The proof of this is found, not in the laws of geometry, but in the study of natural science. I refer to that rare and elusive reptile known to the Indians as Wig-Lum-No-See-Um, which translates as "The Snake That Has Never Been Seen". As all Indians know, when this snake is in danger of being observed, it grabs its tail in its mouth and swallows itself, thereby becoming invisible. When the danger has passed, it spits itself out and again becomes visible but unseen. For this reason, it has never been described by biologists.

Now, my hypothesis was that a 4-D object could be created by the rotation of a 3-D object. A snake is obviously a three-dimensional object, and it must obviously rotate in order to swallow itself. Since this causes it to become invisible, we may safely assume that all four-dimensional objects become invisible while being created.

TIGERCUB ANNOUNCEMENT

Tigercub Software has recently published TI-PD Catalog #2, listing 309 disks of public domain and fairware. The catalog is available by mail, for $1, refundable on the first order - or can be picked up at a C.O.N.N.I. meeting for fifty cents.

Those who have ordered from the first catalog by mail will have received a supplement listing all additions and updates. This supplement is available on request or can be picked up at any C.O.N.N.I. meeting.

The TI-PD disks are sold for just $1.50 per disk, postpaid on orders for 8 or more disks. If ordered for pickup at a C.O.N.N.I. meeting, to save packing and postage, they are just $1.25 per disk.

NOTE - This is NOT the same as Jim Peterson's 40-page public domain catalog, from which he offered to copy programs for .50 each to help C.O.N.N.I. get over a financial crunch.

TIGERCUB
Jim Peterson
156 Collingwood Ave
Columbus, OH 43213

TI—BASE - From INSCEBOT
TUTORIAL 10.1.1 By Martin Smoley
NorthCoast 99'ers - June 17, 1989

Copyright 1989 By Martin A. Smoley

I am reserving the copyright on this material, but I will allow the copying of this material by anyone under the following conditions. (1) It must be copied in its entirety with no changes. (2) If it is retyped, credit must be given to myself and the NorthCoast 99ers, as above. (3) The last major condition is that there may not be any profit directly involved in the copying or transfer of this material. In other words, Clubs can use it in their newsletters and you can give a copy to your friend as long as its free.

This tutorial (I hope) will give you some ideas on how to use the DATE or, D-type fields. Not too long ago I was under the impression that you couldn't do much with a Date type field. Of course I knew you could sort a date field, and find records through the date field, but I didn't think past that point. Then I got into a situation where I needed to calculate the age of a ewe. That's right, I said a ewe, the mother of a lamb. I couldn't do a thing with D fields using MM, DD or YY, so I called Dennis for some help, and did I ever feel like a dummy. Dennis said you can extract a month by using MONTH, a day by using DAY and a year by using YEAR. "Ask the right question and you'll get the right answer (I thought)." Not from Dennis, from TI-Base. Now I'll try to explain it to you. NOTE: In the CFs included in this article the lines with numbers are actual CF or prorgam lines and the lines starting with REC or 0000 are the product of the DISPLAY commands that immediately preceded them. If you are entering these CFs, enter only the lines that have line numbers, but do not enter the line numbers. The (nnn) means refer to a specific line number.

A date field is used to store a date in the form "MM/DD/YY". "I'm sure that this is old news for most of you." This form can be used in a Db field, or in a local variable. An example would be LOCAL BORN D 8 (005). TIB then creates a local variable space 8 units in length under the name BORN and the designation D Type. D type enables TIB to perform Date operations on whatever data it finds stored in that field or variable. BORN is presently empty. REPLACE BORN WITH "02/12/43" (012) would store the date February 12, 1943 in BORN in it's proper form (MM/DD/YY). From this point there are many things that TIB can do with the data stored in BORN. If you enter LOCAL MO N 3 (007) and then REPLACE MO WITH MONTH(BORN) (015), TIB would extract 02 from BORN and place a copy of it in MO. The command DAY(BORN) would extract 12 and YEAR(BORN) would extract 43. If you created AGE N 3 and the current date "06/14/89" was in a date field named CURDT, REPLACE AGE WITH YEAR(CURDT) - YEAR(BORN) (020) would place 46 in AGE. Unfortunately it doesn't work in the other direction. REPLACE YEAR(BORN) WITH "45" does not work (as far as I can tell). If you enter REPLACE BORN WITH "45" (023), the 45 will be place in the far left portion of BORN, which is the month area. DISPLAY BORN would then produce (45   ), without the parenthesis. Concatenation (!) can be used to get the results you want as far as placing data into a date field (025). You should notice that the second "/" was eliminated (024) to allow for the right most space in AGE (004) which has a length of 3.

```
001  CLEAR
002  CLOSE ALL
003  CLEAR LOCAL
004  LOCAL AGE N 3
005  LOCAL BORN D 8
006  LOCAL CURDT D 8
007  LOCAL MO N 3
008  LOCAL DY N 3
009  LOCAL YR N 3
010  LOCAL TEST N 6
011  DISPLAY BORN,CURDT,AGE
-----------------------------------------------
REC   BORN        CURDT       AGE
0000
-----------------------------------------------
012  REPLACE BORN WITH "02/12/43"
013  REPLACE CURDT WITH "06/18/89"
014  DISPLAY BORN,CURDT,AGE
-----------------------------------------------
REC   BORN        CURDT       AGE
0000  02/12/43    06/18/89
-----------------------------------------------
015  REPLACE MO WITH MONTH(BORN)
016  REPLACE DY WITH DAY(BORN)
017  REPLACE YR WITH YEAR(BORN)
018  DISPLAY MO,DY,YR,AGE
-----------------------------------------------
REC   MO    DY    YR    AGE
0000   2    12    43
-----------------------------------------------
019  REPLACE AGE WITH YEAR(CURDT);
020                   - YEAR(BORN)
021  DISPLAY BORN,CURDT,AGE,TEST
-----------------------------------------------
REC   BORN        CURDT       AGE   TEST
0000  02/12/43    06/18/89     46
-----------------------------------------------
022  REPLACE BORN WITH "45"
023  DISPLAY BORN,CURDT,AGE,TEST
-----------------------------------------------
REC   BORN        CURDT       AGE   TEST
0000  45          06/18/89     46
-----------------------------------------------
024  REPLACE CURDT WITH "00/00" ! AGE
025  DISPLAY BORN,CURDT,AGE,TEST
-----------------------------------------------
REC   BORN        CURDT       AGE   TEST
0000  45          00/00 46     46
-----------------------------------------------
026  REPLACE BORN WITH "06/31/44"
027  REPLACE TEST WITH DAY(BORN);
                     * MONTH(BORN)
028  DISPLAY BORN,CURDT,AGE,TEST
-----------------------------------------------
REC   BORN        CURDT       AGE   TEST
0000  06/31/44    00/00 46     46        186
-----------------------------------------------
029  RETURN Copyright Martin A. Smoley
030  *                               1989
031  *                     TDT3/C
```

Continued Next Page.

TI-BASE - From INSCEBOT
TUTORIAL 10.1.2 By Martin Smoley

NorthCoast 99'ers - June 17, 1989

After you become familiar with their basic functions you can do some interesting things with Date type fields, but you must be careful to remember what you have put in a field and what its position is. For example, REPLACE DATE WITH "Ma/rt/in" will place the character string Ma/rt/in into DATE. You could store Ma/rt/in in a DB and retreive it later. However, TIB will recognize these as Characters and *will not allow you to do anything with them. If you could* multiply Ma by 2.2, I can't imagine what the result might be. But on the other hand, TIB seems to be able to recognize numbers which have been entered as characters and then place them into a date type field. I have attempted to show the different aspects of this theory in TDT5 and TDT6. Notice that DATEC is a character variable (006). I then placed "12345.78" into DATEC as a character string (012). The reason I did it in this manner was to make sure that TIB considered 12345.78 to be characters at this point, even though it looks like a number. I then transfered DATEC to *DATE (013). DATE is a D or Date type field (005).* I was then able to do any Date type function involving DATE, including multiplication of its parts (019), or multiplication by a constant, or LITERAL, (020). Then I went through the same steps using a numeric (N) type field. Notice in line # 24 that TMP (004) is a numeric field and that 77.77 has no quotes around it. I am attempting to guarantee that 77.77 is a number. Then I placed TMP into DATE (026) and performed the Date type functions on 77.77. The reason I added the zeros was to demonstrate what I said previously, that you need to remember the exact position of *the data in a field if you expect to use it in this manner.* TMP has a width of 8 (004). This means it would fit right into a Date type field. When I added the "00", I moved the spacing to put "7." in the year portion of DATE. Check the " / / " spacing to see where the slashes are located. This means, when YR is multiplied by 100 (032), TIB is multiplying 7 by 100. If you work with this idea, understand it and are very careful, you could use the date type field to extract specific parts of a number. A very simple demonstration of this would be to place a dollar and cents type number into a date field with the cents portion in the year segment of the date field. *You could then use REPLACE* CENTS WITH YR(DATE) to extract the cents, if that's what you need. Then REPLACE dollars WITH dollars - CENTS would give you the whole dollar amount. I realize that in most cases this idea is to cumbersome to use, but if there is an instance when no other procedure will work, this idea just might do the trick.

SORT by INSCEBOT

I don't know if SORT is the name that will be used, but sort is what it does. I am currently testing this new sort program *from those TI-Base guys and it looks great. It will* sort TI-Base files, Fix file, Var files and hopefully soon Basic display files. I've tried it on TIB files and it's real fast and easy to use. It should be available soon as a separate disk for under $15.00 (I think). The fact that it works on many different file types will make it a very useful [illegible]

I have pulled TDT5 and TDT6 together to save space. I hope this does not confuse everyone.

```
001  CLEAR
002  CLOSE ALL
003  CLEAR LOCAL
004  LOCAL TMP N 8 2
005  LOCAL DATE D 8
006  LOCAL DATEC C 8
007  LOCAL MO N 3
008  LOCAL DY N 3
009  LOCAL YR N 3
010  LOCAL TEST N 12 2
011  ***********************************
012  REPLACE DATEC WITH "12345.78"
013  REPLACE DATE WITH DATEC
014  DISPLAY DATE,TMP,DATEC
-----------------------------------------
REC     DATE        TMP         DATEC
0000    12345.78                12345.78
-----------------------------------------
015  REPLACE MO WITH MONTH(DATE)
016  REPLACE DY WITH DAY(DATE)
017  REPLACE YR WITH YEAR(DATE)
018  DISPLAY DATE,MO,DY,YR,TEST
-----------------------------------------
REC     DATE        MO    DY    YR    TEST
0000    12345.78    12    45    78
-----------------------------------------
019  REPLACE TEST WITH DY * YR
020  REPLACE TMP WITH YR * 2.2
021  DISPLAY DATE,TMP,TEST
-----------------------------------------
REC     DATE        TMP         TEST
0000    12345.78    171.60           3510.00
-----------------------------------------
022  *                               TDT5/C
023  ***********************************
024  REPLACE TMP WITH 77.77
025  DISPLAY DATE,TMP,TEST
-----------------------------------------
REC     DATE        TMP         TEST
0000    12345.78     77.77           3510.00
-----------------------------------------
026  REPLACE DATE WITH "00" ! TMP
027  REPLACE MO WITH MONTH(DATE)
028  REPLACE DY WITH DAY(DATE)
029  REPLACE YR WITH YEAR(DATE)
030  DISPLAY DATE,MO,DY,YR,TMP
-----------------------------------------
REC     DATE        MO    DY    YR    TMP
0000    00   77.    0     0     7        77.77
>>==>>   /   /    <<==<< Note Spacing
-----------------------------------------
031  *
032  REPLACE TEST WITH YR * 100
033  DISPLAY DATE,YR,TEST
-----------------------------------------
REC     DATE        YR    TEST
0000    00   77.    7               700.00
-----------------------------------------
034  RETURN  Copyright Martin A. Smoley
035  *                                1989
036  *                         TDT6/C
```

MEETING DATES FOR 1989 - 90

3RD SATURDAY
16 DEC 1989
20 JAN 1990
17 FEB 1990
17 MAR 1990
21 APR 1990
19 MAY 1990
16 JUN 1990
21 JUL 1990
18 AUG 1990
15 SEP 1990
20 OCT 1990
17 NOV 1990

4TH WEDNESDAY
27 DEC 1989
24 JAN 1990
28 FEB 1990
28 MAR 1990
25 APR 1990
23 MAY 1990

C.O.N.N.I. BOARD MEMBERS

Pres. - Dick Beery (614) 459-3597
Vice Pres. - Jim Seitz (614) 875-5532
Treas. - *(acting) D.Beery/E.Wade*
Sec. - Charles Osment (614) 833-1698
Membership - Everett Wade (614) 262-6346
Librarian - Chuck Grimes (614) 268-8821
Cassette - Sonny Grubb (614)1-873-8708
Cartridges - Dave Truesdale (614) 238-0719
NL Exchange -Curt Borders (614) 279-5208
TIBBS - Irwin Hott (614) 263-5319
Dick Beery (614) 459-3597
Editor- Jean Hall (614) 885-4223
Assist.- Carole Parkins (614) 891-4965

****
NEW MEETING PLACE
CHEMICAL ABSTRACTS
2540 OLENTANGY RD
****

Spirit of 99

Bulk Rate
U.S. Postage
PAID
COLUMBUS 43221
Permit No. 1945

C.O.N.N.I.
181 HEISCHMAN AVE
WORTHINGTON, OH 43085

TIME SENSITIVE MATERIAL
POSTMASTER - PLEASE DELIVER PROMPTLY

EXCHANGE NL

SMAUG USER'S GROUP/99
RT 4, BOX 23
BREWTON, AL 36426

*** MEMBERSHIP APPLICATION ***

NAME____AGE______
ADDRESS____
CITY_____________________________STATE________ZIP________
AREA CODE______HOME PHONE______________BUSINESS PHONE_____________EXT#_____
WHAT IS YOUR PROFESSION/VOCATION_________________________
HOW LONG HAVE YOU OWNED YOUR COMPUTER?_____________________
DATE OF APPLICATION_________________ACCEPTED BY_______________